



**KERNFORSCHUNGSANLAGE JÜLICH GmbH**

Zentrallabor für Elektronik  
Institut für Festkörperforschung

**RECHNERGESTEUERTE ANLAGE ZUR  
UNTERSUCHUNG VON EINKRISTALLEN  
AM GAMMADIFFRAKTOMETER**

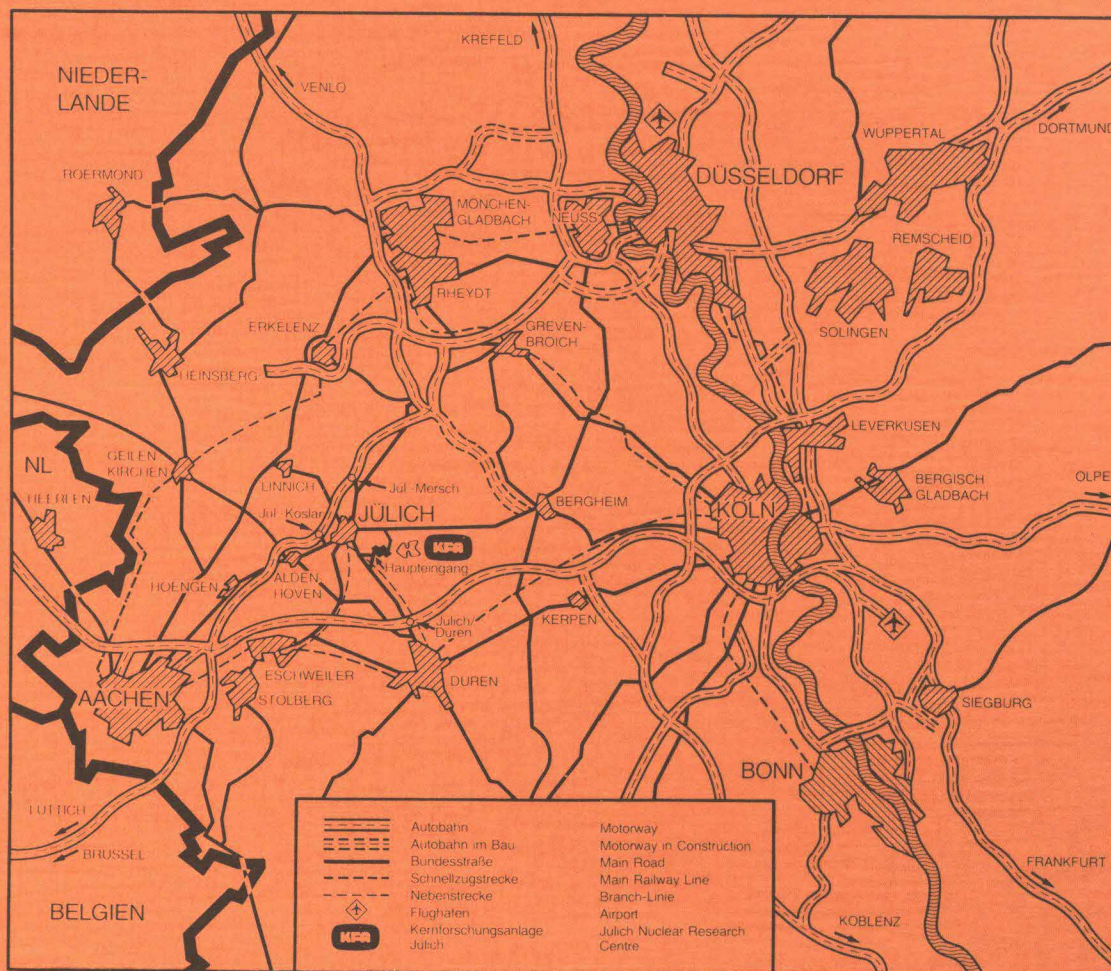
**G A D I**

von

H. Heer, H. Bleichert, W. Gruhn, R. Möller

Jül - Spez - 277  
Oktober 1984  
ISSN 0343-7639





Als Manuskript gedruckt

**Spezielle Berichte der Kernforschungsanlage Jülich – Nr. 277**

Zentrallabor für Elektronik · Institut für Festkörperforschung Jül-Spez-277

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH  
Postfach 1913 · D-5170 Jülich (Bundesrepublik Deutschland)  
Telefon: 02461/610 · Telex: 833556-0 kf d

**RECHNERGESTEUERTE ANLAGE ZUR  
UNTERSUCHUNG VON EINKRISTALLEN  
AM GAMMADIFFRAKTOMETER**

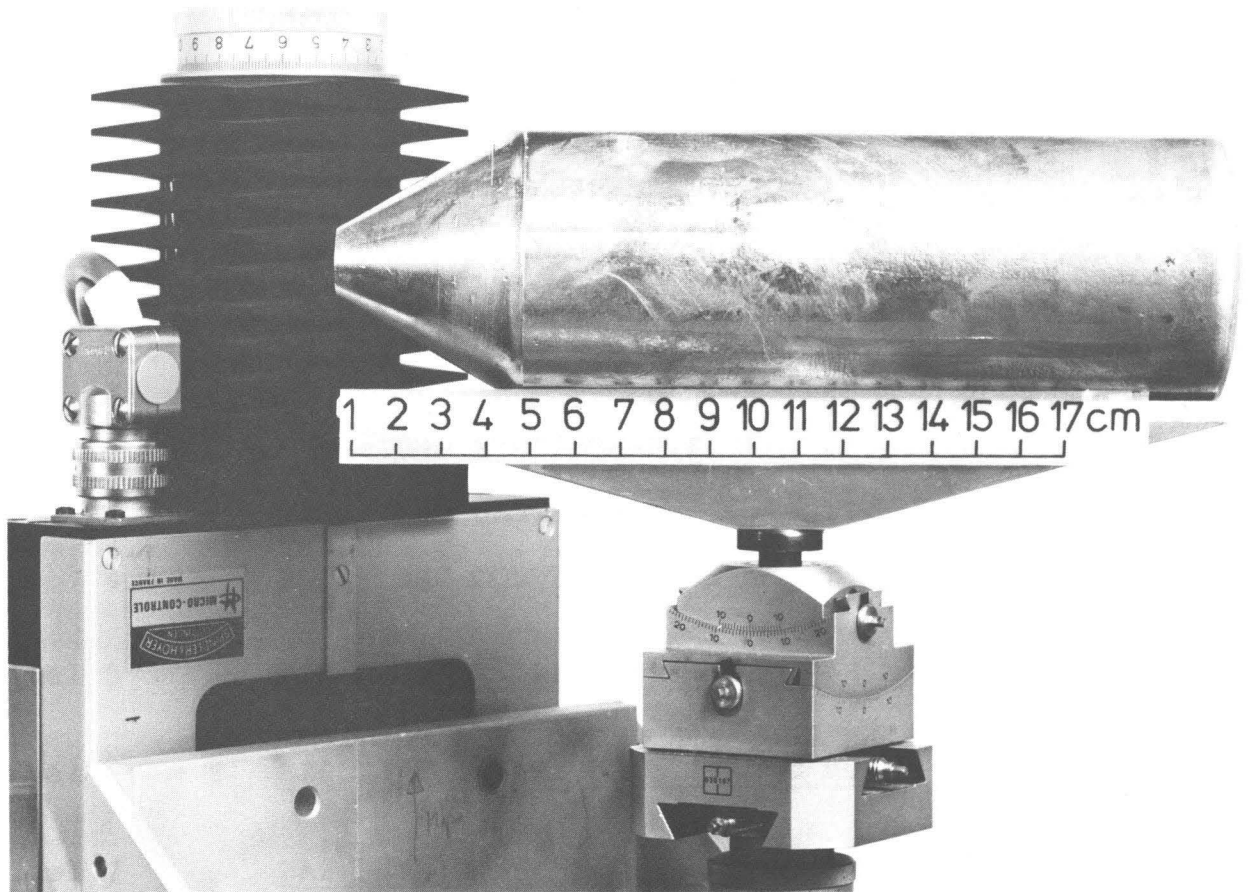
**G A D I**

von

H. Heer, H. Bleichert, W. Gruhn, R. Möller







Das Bild zeigt einen großformatigen Kupfereinkristall zur Herstellung eines Neutronenmonochromators



## Inhaltsverzeichnis

Kapitel 1	<u>ZIELSETZUNG</u>	
Kapitel 2	<u>EINLEITUNG</u>	
2.1	VORBEMERKUNGEN . . . . .	2
2.2	PHYSIKALISCHER HINTERGRUND . . . . .	3
Kapitel 3	<u>AUFBAU DER MESSAPPARATUR</u>	
3.1	DIE STRAHLUNGSQUELLE . . . . .	6
3.2	DIE KOLLIMATION . . . . .	7
3.3	DIE DREHVORRICHTUNG . . . . .	7
3.4	DER STRAHLUNGSDETEKTOR . . . . .	7
Kapitel 4	<u>MESS- UND STEUERGERÄTE</u>	
4.1	SCHRITTMOTORSTEUERUNG . . . . .	9
4.2	ZÄHLER . . . . .	11
4.3	ADC - DAC CONVERTER . . . . .	12
Kapitel 5	<u>DIE RECHNERKONFIGURATION</u>	
Kapitel 6	<u>EXPERIMENT - RECHNER I**2 C BUS</u>	
Kapitel 7	<u>DAS BETRIEBSSYSTEM RSX11M 4.0</u>	
7.1	DIE SYSTEMGENERIERUNG . . . . .	16
7.2	STARTUP - AKTIVITÄTEN . . . . .	17
Kapitel 8	<u>SOFTWARE - ENTWICKLUNGEN</u>	
8.1	ENTWICKLER SEITE . . . . .	18
8.2	ANWENDER SEITE . . . . .	20
Kapitel 9	<u>SOFTWARE</u>	
9.1	I**2 C-BUS SOFTWARE . . . . .	22
9.2	MOTORSTEUERUNG . . . . .	24
9.3	ZÄHLER ANSTEUERUNG . . . . .	26
9.4	ADC - DAC - ANSTEUERUNG . . . . .	28
9.5	DISPLAY - ANSTEUERUNG . . . . .	30
9.5.1	Unterprogramm LADA . . . . .	30
9.5.2	Unterprogramm CODA . . . . .	33

9.5.3	Unterprogramm BACO . . . . .	34
9.5.4	Unterprogramm AXDA . . . . .	35
9.6	<b>ALLGEMEINE UNTERPROGRAMME . . . . .</b>	<b>36</b>
9.6.1	Dialog mit dem Anwender über die Experimentparameter . . . . .	36
9.6.2	Ja / Nein Beantwortung . . . . .	39
9.6.3	Initialisierung des Counters im Scaler-Modul . .	40
9.6.4	Mitteilungen an den Anwender . . . . .	41
9.6.5	Aufnahme von Zählraten . . . . .	42
 Kapitel 10	 <u><b>DER MESSABLAUF</b></u>	
10.1	<b>DIE HANDJUSTIERUNG . . . . .</b>	<b>43</b>
10.2	<b>DIE AUFNAHME VON ROCKINGKURVEN . . . . .</b>	<b>43</b>
10.3	<b>DER INITIALISIERUNGS-DIALOG . . . . .</b>	<b>46</b>
 Kapitel 11	 <u><b>DIREKT ACCESS FILES</b></u>	
11.1	<b>ZWECK . . . . .</b>	<b>48</b>
11.2	<b>DEKLARATION . . . . .</b>	<b>48</b>
11.3	<b>DATENFORMAT . . . . .</b>	<b>49</b>
11.4	<b>BEARBEITUNG DES DA-FILES . . . . .</b>	<b>50</b>
 Kapitel 12	 <u><b>DIE AUSWERTUNG DER MESSDATEN</b></u>	
 Kapitel 13	 <u><b>TESTSOFTWARE FÜR I**2 C BUS MODULE</b></u>	
13.1	<b>MOTORTEST ( TEMO ) . . . . .</b>	<b>55</b>
13.2	<b>ZÄHLERTEST ( COTE ) . . . . .</b>	<b>56</b>
13.2.1	<b>ADC / DAC - Modultest ( ADCT ) . . . . .</b>	<b>57</b>
 Kapitel 14	 <u><b>DIE FILE-ORGANISATION</b></u>	
14.1	<b>DIE WINCHESTER PLATTE . . . . .</b>	<b>58</b>
 Kapitel 15	 <u><b>AUSBLICK</b></u>	
 Kapitel 16	 <u><b>FOTOGRAPHIEN</b></u>	
 Kapitel 17	 <u><b>DANKSAGUNG</b></u>	
 Kapitel 18	 <u><b>LITERATUR</b></u>	



## Kapitel 1

### ZIELSETZUNG

Bei der Untersuchung von Festkörpereigenschaften ist es in vielen Fällen notwendig, mit Einkristallen zu arbeiten. Der gestiegene Bedarf in der Industrie und Forschung, sowie der Wunsch nach besserer Charakterisierung mittels  $\gamma$ -Diffraktometrie machten es notwendig, das bestehende Instrument auszubauen und zu modernisieren.

Die Vorteile einer rechnergesteuerten Anlage gegenüber dem herkömmlichen, halbautomatischen Betrieb sind vielfältig. Nicht nur die Prozesssteuerung, sondern auch die Datenerfassung und Auswertung werden vom Rechner übernommen. Durch eine Fernbedienung ist der Bediener in der Lage, schnell einen Reflex zu finden und einen Kristall in jede gewünschte Messposition zu fahren. Die lückenlose Protokollierung aller wichtigen Messparameter, die bequeme Datenspeicherung, sowie die automatische Auswertung sind für den Anwender von grossem Nutzen. Schliesslich kann die Messzeit auf praktisch 24 Stunden pro Tag gesteigert werden. Damit wird die routinemässige Charakterisierung mittels  $\gamma$ -Diffraktometrie auf ein völlig neues Niveau gestellt.

## Kapitel 2

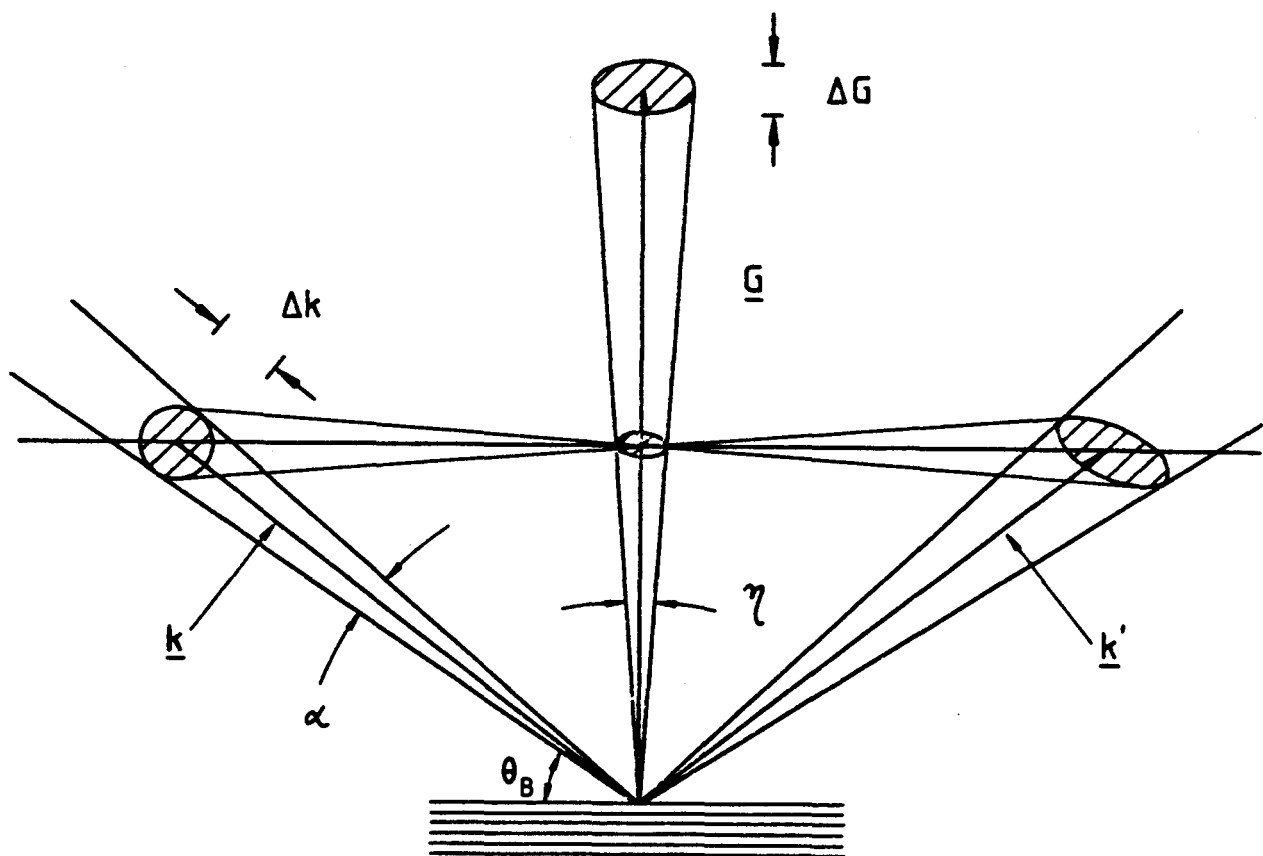
### EINLEITUNG

#### 2.1 VORBEMERKUNGEN

Die Gamma-Diffraktometrie ist seit 10 Jahren zu einem festen Bestandteil bei der Charakterisierung von Einkristallen im Institut für Materialentwicklung im IFF geworden. Das Messprinzip wurde ursprünglich von H. Maier-Leibnitz 1969 vorgeschlagen, um die Mosaikstruktur von Monochromator-Kristallen für Neutronenspektrometer am Höchstflussreaktor in Grenoble zu bestimmen. Heute reicht das Einsatzgebiet des Gamma Diffraktometers im IFF von Untersuchungen an wasserstoffbeladenen Niobeinkristallen bis zu Turbinenschaufeln aus mehrkomponentigen Legierungen. Das prominenteste Untersuchungsobjekt war jüngst ein Siliziumkristall, der im Weltraumlabor SPACELAB gezüchtet wurde.

## 2.2 PHYSIKALISCHER HINTERGRUND

Streuexperimente von elektromagnetischer Strahlung am Festkörper liefern uns Informationen über die atomare Kristallstruktur. Dabei muss die Wellenlänge  $\lambda$  der untersuchenden Strahlung mindestens im Bereich interatomarer Abstände  $d_{hkl}$  liegen. Man erhält genau dann in eine bestimmte Richtung gestreute Strahlung, wenn die BRAGG-Bedingung erfüllt ist :  $\underline{k}' - \underline{k} = \underline{G}$ .



**Fig. 1: Bragg-Bedingung im reziproken Raum**

- $\underline{k}$ : Wellenzahlvektor der einfallenden Strahlung,
- $\underline{k}'$ : Wellenzahlvektor der reflektierten Strahlung,
- $\underline{G}$ : Reziproker Gittervektor,
- $\alpha$ : Divergenz der einfallenden Strahlung
- $\eta$ : Mosaikwinkel des Kristalls

Im realen Kristall hat man sowohl Schwankungen in der Länge von  $G$  (die atomaren Netzebenen haben nicht überall den gleichen Abstand), als auch in der Richtung von  $G$  (die Netzebenen sind nicht parallel). Die Abweichung von der Vorzugsrichtung des Kristalls ist durch den Mosaikwinkel  $\eta$  gekennzeichnet. Dreht man den Kristall um eine Achse, die senkrecht zur Zeichenebene in Fig.1 steht, in einem kleinen Winkelbereich um den BRAGG-Winkel  $\theta_B$ , so kommen nacheinander die (verkippten) Netzebenen zur Reflexion. Trägt man die gebeugte Intensität über dem Drehwinkel auf, erhält man eine sogenannte Rockingkurve. Diese spiegelt die Verteilungsfunktion (Mosaikverteilung) der Verkipnungen wieder. Die gemessene Reflektivitätsverteilungsfunktion  $r_m(\omega)$  ist gegeben durch :

$$r_m(\omega) = \frac{I_B(\omega) - I_U}{I_P}$$

$I_B(\omega)$  : Intensität des reflektierten Strahls  
 $I_U$  : Untergrundintensität  
 $I_P$  : Primärstrahlintensität ausserhalb des Braggreflexes

Da die oben aufgeführten Intensitäten der gebeugten Strahlung alle durch die Probe gemessen werden und sich das durchstrahlte Volumen nicht wesentlich ändert, ist die Kenntnis des Absorptionskoeffizienten nicht notwendig. Das Integral  $R_m = \int r(\omega) d\omega$  bezeichnet man als das gemessene integrale Reflexionsvermögen, das eine reine Kristalleigenschaft wiedergibt und nicht etwa von der einfallenden Intensität abhängt.

Im Gegensatz zu Röntgenstrahlen ( $\lambda$  einige  $10^{-10}$  m) misst man mit Gammastrahlen ( $\lambda$  einige  $10^{-12}$  m) im wesentlichen Netzebenenverkipnungen. Dabei erlaubt das hohe Durchdringungsvermögen der Gamma-Strahlung Messungen an einkristallinen Proben von einigen Zentimetern Dicke. Die Kenntnis der Mosaikverteilung eines Einkristalls gibt dem Experimentator wertvolle Informationen darüber, ob der Kristall für weitergehende Messungen oder Anwendungen geeignet ist.



## Kapitel 3

### AUFBAU DER MESSAPPARATUR

Das verwendete Gamma-Diffraktometer ist ein Ein-Kristallinstrument, bei dem sich nur der zu untersuchende Kristall zwischen Strahlungsquelle und Strahlungsdetektor befindet. Das Instrument setzt sich zusammen aus Gammaquelle, Kollimator, Goniometer mit Probe und einem in horizontaler Richtung verschiebbaren NaJ-Szintillationszähler (Fig. 2).

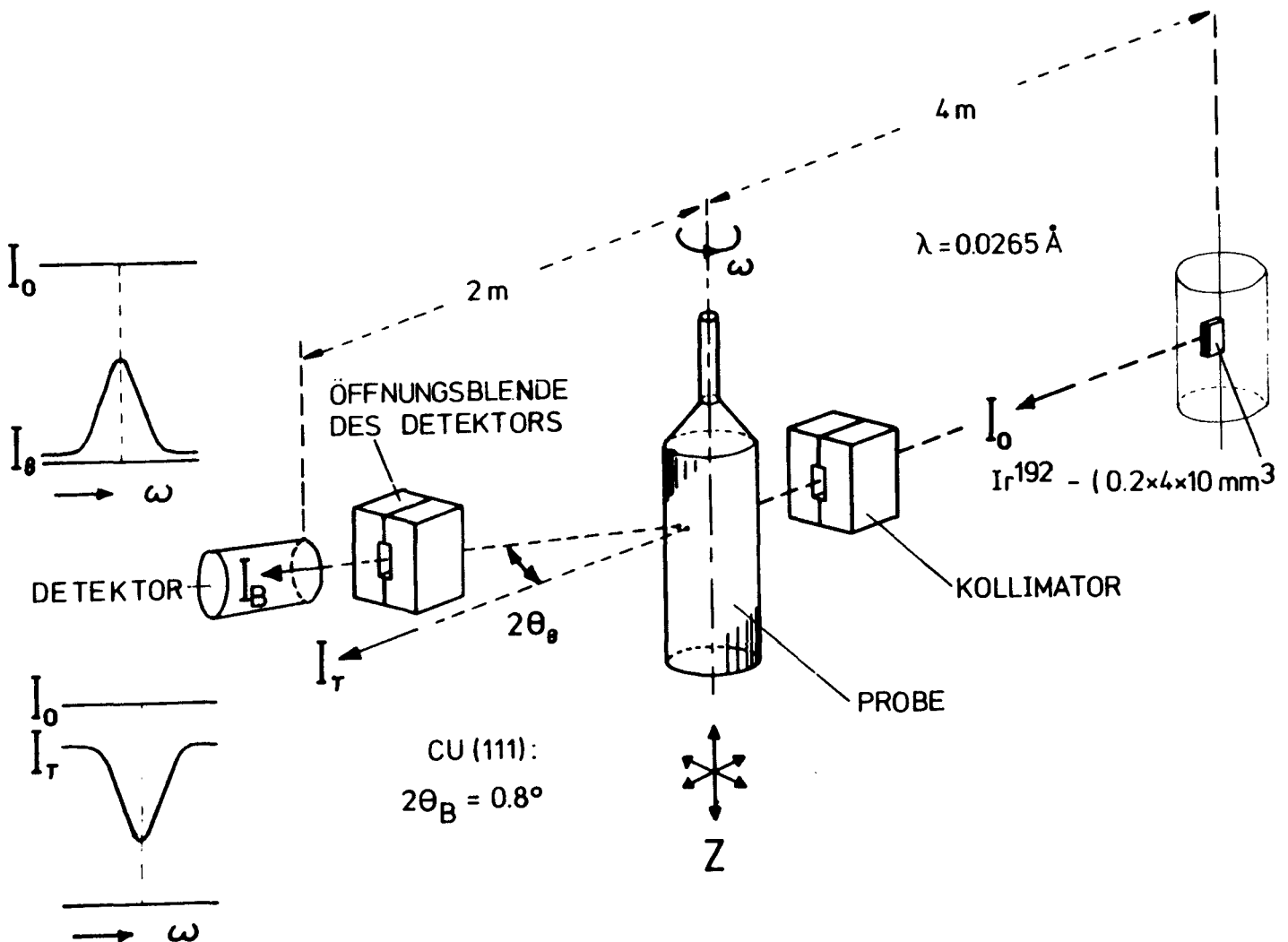


Fig. 2: Prinzipskizze des Gammadiffraktometers

### 3.1 DIE STRAHLUNGSQUELLE

Die benutzte Gamma-Quelle besteht aus einem Iridiumblech (0.2\*4\*10mm\*\*3), das in einer Bleiabschirmung so positioniert ist, dass nur Strahlung in Richtung der Probe austreten kann. Als Ausgangsmaterial wird natürliches Iridium mit folgender Isotopenzusammensetzung verwendet :

$$38.5\% \text{ } ^{77}\text{Ir}^{191}$$

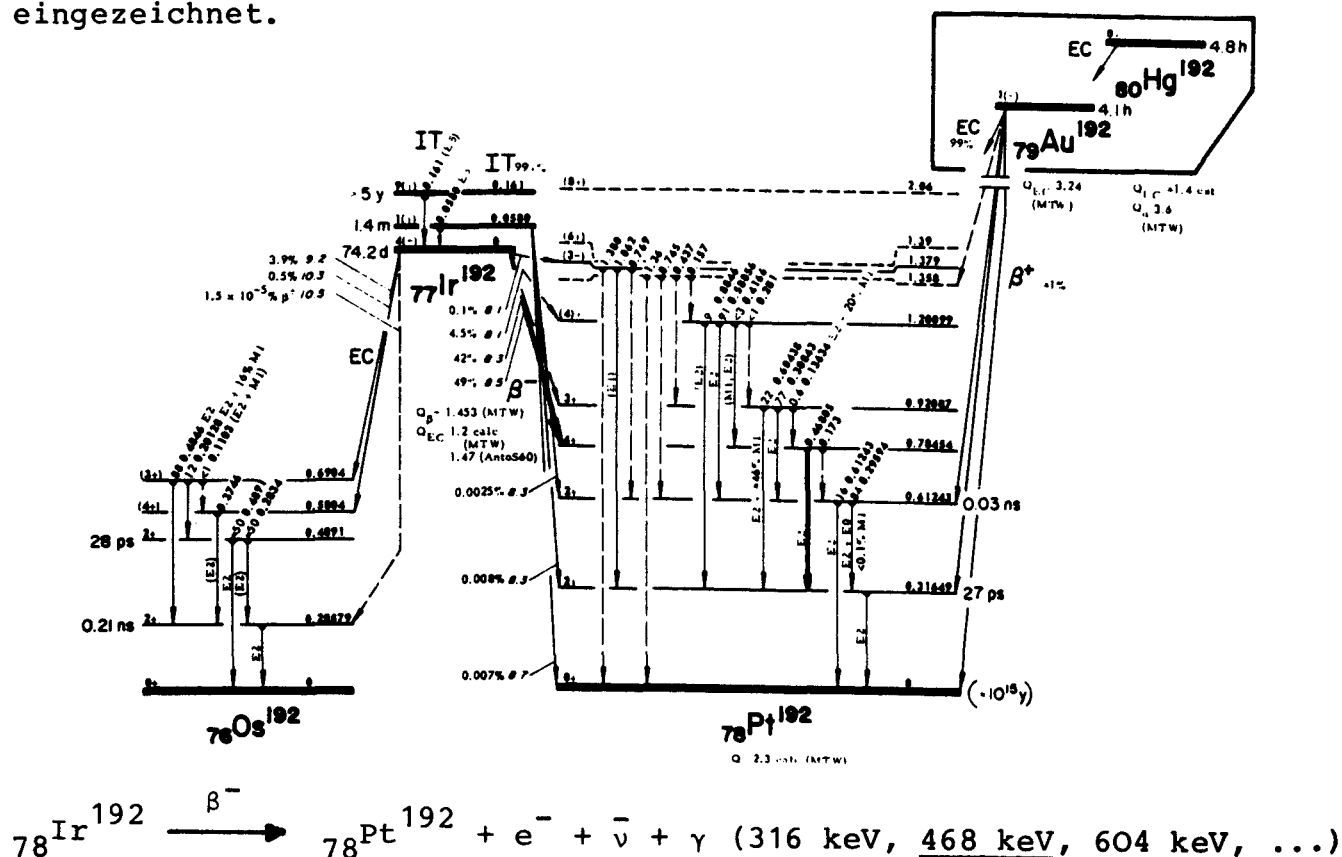
$$61.5\% \text{ } ^{77}\text{Ir}^{193}$$

Durch Bestrahlung im Jülicher Forschungsreaktor MERLIN (Neutronenfluss  $10^{14} \text{ n cm}^{-2} \text{ sec}^{-1}$ ) entstehen daraus die radioaktiven Isotope :

$$^{77}\text{Ir}^{192} \quad \text{Halbwertszeit 74 Tage}$$

$$^{77}\text{Ir}^{194} \quad \text{Halbwertszeit 19 Tage}$$

Nach Installation am Gamma-Diffraktometer beträgt die Aktivität des Bleches maximal 100 Ci. Wegen der grossen Halbwertszeit des  $\text{Ir}^{192}$  wird die Quelle nur etwa alle 4 Monate ausgewechselt. Die zur Messung benötigten  $\gamma$ -Linien werden durch den Zerfall von  $\text{Ir}^{192}$  in angeregte Zustände des Zerfallproduktes  $^{78}\text{Pt}^{192}$  geliefert. Die Energie der emittierten Photonen ist gleich der Differenz der entsprechenden Energieniveaus. In Fig.3 ist das Kerntermschema des hier wichtigen Platinisotopes dargestellt. Der Hauptzerfallskanal ist dick eingezeichnet.



### 3.2 DIE KOLLIMATION

Durch eine Energieunschärfe von  $\Delta E/E = 10^{-6}$  ist die Strahlung ausreichend monochromatisch und es entfällt ein Monochromatorkristall. Die Divergenz der Strahlung wird einerseits durch die Stirnfläche des Ir-Blechtes, andererseits durch Bleiklötze mit festgelegten Schlitzbreiten definiert. Mit der Breite von 0.2 mm des Ir-Blechtes, einem Abstand Quelle-Bleikollimator von ca. 4 m und einer Schlitzbreite von 0.2 mm erhält man eine beste Divergenz von 10 Winkelsekunden.

### 3.3 DIE DREHVORRICHTUNG

Zur Einstellung des BRAGG-Winkels befindet sich die zu untersuchende Probe auf einem Goniometer, welches mit 4 Schrittmotoren bewegt werden kann. Man hat so die Möglichkeit die Probe horizontal und vertikal zu verschieben und kann auf diese Weise jede Stelle im Kristall untersuchen. Die Aufnahme der Rockingkurve selber geschieht wahlweise mit einer Grob- oder Feinrotation. Die Grobrotation ermöglicht den Kristall um 360 Grad zu drehen mit einer kleinsten Rotationseinheit von 7 Winkelsekunden. Die Feinrotation überstreicht einen Winkelbereich von 6 Grad mit einer kleinsten Rotationseinheit von 2 Winkelsekunden.

### 3.4 DER STRAHLUNGSDETEKTOR

Zum Nachweis der gebeugten Strahlung wird ein NaJ-Szintillationszähler eingesetzt. Die auf den Detektor auftreffenden Gamma-Quanten wechselwirken mit den Elektronen des NaJ-Kristalls hauptsächlich durch Photo- und Comptoneffekt. Die dabei auftretenden Lichtblitze (Szintillationen) fallen auf eine Photokathode und lösen dort Photoelektronen aus. Diese werden in einem Sekundärelektronenvervielfacher (S.E.V.) verstärkt und in einer nachgeschalteten Elektronik in einen Spannungspuls umgewandelt. Dabei ist die Amplitude des Pulses proportional der vom Szintillator absorbierten Gamma-Strahlungsenergie. Die Impulse laufen dann über einen Impulshöhendiskriminator (Einkanalanalysator), der nur solche Impulse weitergibt, deren Höhe in einem vorgegebenen Intervall (Window) über einer vorgegebenen Schwelle (Baseline) liegen.

Fig. 4 zeigt das Impulshöhenspektrum der Ir-Quelle. Obwohl nur monochromatische Strahlung auf den Detektor trifft, tritt eine Verbreiterung der Linien ein. Diese wird hervorgerufen durch Schwankungen der Lichtzuführung Nachweiskristall-Photokathode und durch mangelhafte elektronenoptische Fokussierung im S.E.V.. Wegen des schlechten Auflösungsvermögens des NaJ-Detektors ( $\Delta E/E$  10% bei 468 keV) ist es nicht möglich die intensitätsreichere 316 keV-Linie zu nutzen, ohne dass die benachbarten Energien (308 keV, 295 keV) zum Nachweis gelangen. Das Diskriminatorfenster wird schliesslich so gesetzt, dass nur solche Impulshöhen, die der Energie von 468 keV entsprechen, gemessen werden.

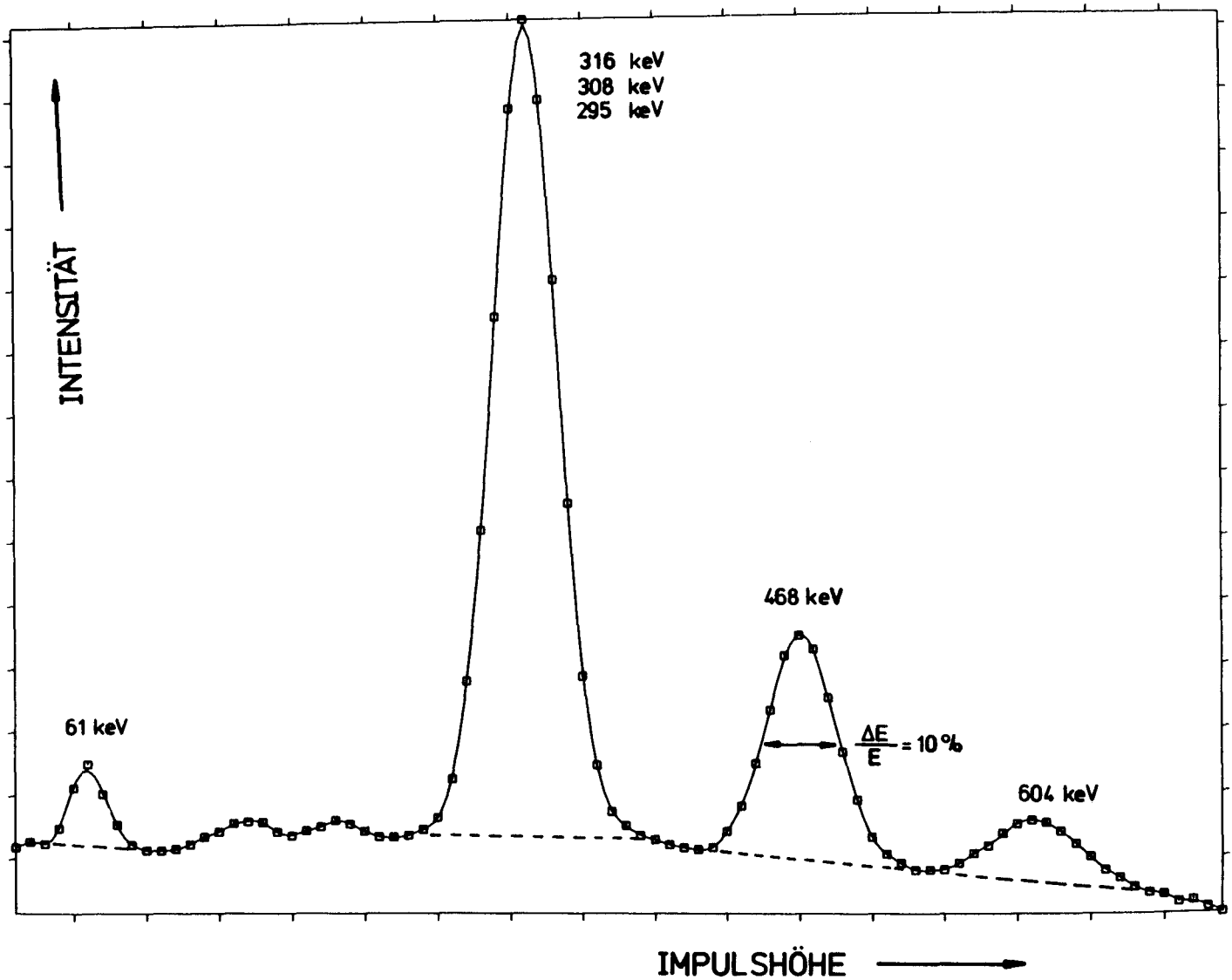


Fig. 4: Impulshöhenspektrum der  $\text{Ir}^{192}$ -Quelle, aufgenommen mit einem NaJ-Szintillationszähler.



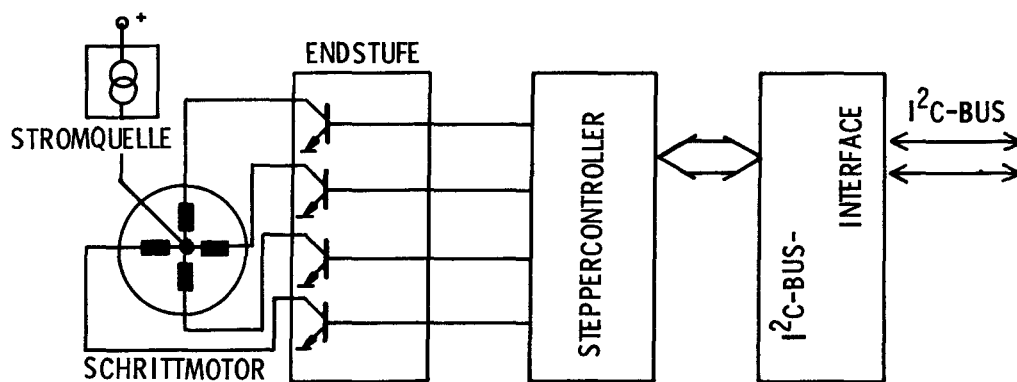
## Kapitel 4

### MESS- UND STEUERGERÄTE

#### 4.1 SCHRITTMOTORSTEUERUNG

Wie bereits in Abschnitt 3.3 erläutert, wird das Goniometer mit 4 Schrittmotoren bewegt. Es handelt sich bei diesen um 4-Phasen-Schrittmotoren (Firma: Slo-Syn Type: M062 FC 03), die mit einem Strom von 1 Ampere pro Phase angetrieben werden.

Die Ansteuerung eines Schrittmotors ist im folgenden Blockschaltbild dargestellt:



**Fig. 5:** Blockschaltbild der Steppermotoransteuerung

Der Strom für jeden Schrittmotor wird von einer Konstantstromquelle (KFA-ZEL/NE 452) geliefert und von der Endstufe auf die entsprechenden Motorphasen geschaltet. Drehrichtung und Motorgeschwindigkeit werden durch die Reihenfolge und Frequenz der Impulse bestimmt, die die Endstufentransistoren schalten.

Die Endstufe wird vom Steppercontroller angesteuert, einem hochintegrierten IC (PPMC 101B), der alle dafür erforderlichen Signale zur Verfügung stellt. Über Kommandos an den Steppercontroller können z.B. Drehrichtung, Anzahl der Motorschritte, Anfahr- und Endgeschwindigkeit usw angegeben werden. Ausserdem fragt der Steppercontroller die Endschalter, soweit vorhanden, ab und stoppt den entsprechenden Motor beim Ansprechen eines Endschalters.

Die Kommandos an den Steppercontroller werden vom I\*\*2 C-Bus-Interface vom I\*\*2 C-Bus empfangen, überprüft und an den Steppercontroller weitergegeben und dort ausgeführt. Beim Beginn und nach Beendigung der Ausführung schickt das I\*\*2 C-Bus-Interface Bestätigungsmeldungen auf den I\*\*2 C-Bus.

I\*\*2 C-Bus-Interface, Steppercontroller und Endstufe sind im Steppercontrollermodul (KFA-ZEL/NE 448) untergebracht.

#### 4.2 ZÄHLER

Die vom Impulshöhendiskriminator gelieferten Impulse (siehe Abs. 3.4) werden vom Zählermodul (KFA-ZEL/NE 447) über ein bestimmtes Zeitintervall gezählt. Dadurch erhält man eine der Strahlungsintensität proportionale Zählrate.

Die folgende Abbildung zeigt das Blockschaltbild dieses Zählermoduls.

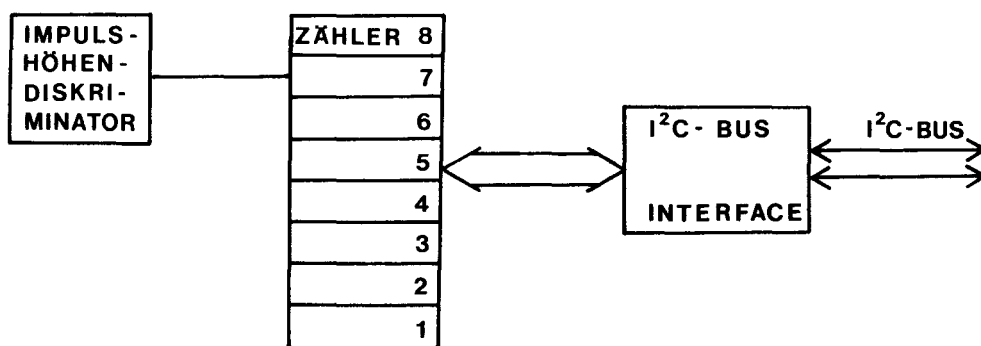
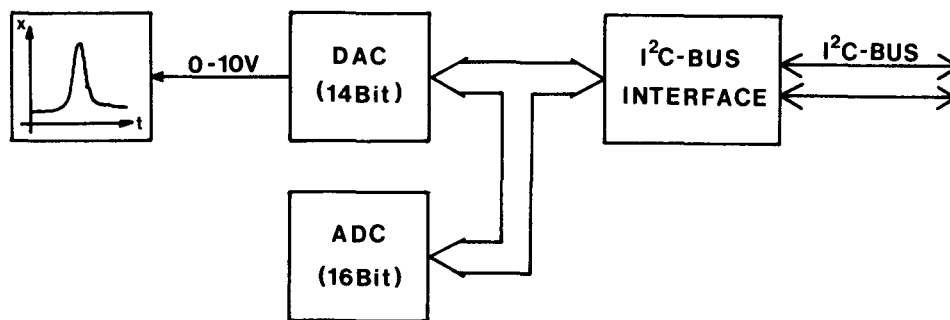


Fig. 6: Blockschaltbild des Zählermoduls

Das Zählermodul besteht aus acht 16Bit-Zählern, von denen allerdings nur zwei Stück (Zähler 7 und 8) zur Strahlungsintensitätsmessung benutzt werden. Dabei dient Zähler 7 zum Zählen der Impulse und Zähler 8 als Zeitgeber. Zähler 8 wird von einem internen Quarzoszillator getaktet und zählt von einem vorsetzbaren Wert rückwärts bis Null. Der Zählerinhalt wird alle 10ms um 1 vermindert, dadurch ergibt sich die Messzeit aus dem vorgesetzten Zählerinhalt x 10ms. Hat der Zähler 8 den Wert Null erreicht, wird Zähler 7 ausgelesen und dessen Inhalt über das I<sup>2</sup>C-Interface und dem I<sup>2</sup>C-Bus zum Rechner geschickt. Dieser Ablauf kann durch ein Kommando über den I<sup>2</sup>C-Bus gestartet werden. Mit diesem Kommando wird gleichzeitig die Messzeit eingestellt. Um die Strahlungsintensität mehrmals hintereinander messen zu können, wiederholt sich der oben beschriebene Vorgang im Zähler automatisch solange, bis er durch ein anderes Kommando gestoppt wird. Die so nacheinander gemessenen Zählraten können im Rechner aufaddiert und anschliessend der Durchschnittswert gebildet werden. Wird beim Starten des Messvorgangs keine neue Messzeit eingestellt, so wird die Messzeit der vorherigen Messung genommen.

#### 4.3 ADC - DAC CONVERTER

Zur graphischen Kontrolle der zu messenden Rockingkurve werden die Zählraten während der laufenden Messung auf einem X-t-Schreiber mitgeschrieben. Dazu muss die vom Zählermodul gelieferte Zählrate, die als digitaler Wert im Rechner vorliegt, in eine analoge Spannung umgewandelt werden. Dies geschieht mit einem Digital-Analog-Converter (DAC), einem Teil eines kombinierten ADC- (Analog-Digital-Converter)-DAC-Moduls (KFA-ZEL/NE 450), das in folgendem Blockschaltbild dargestellt ist.



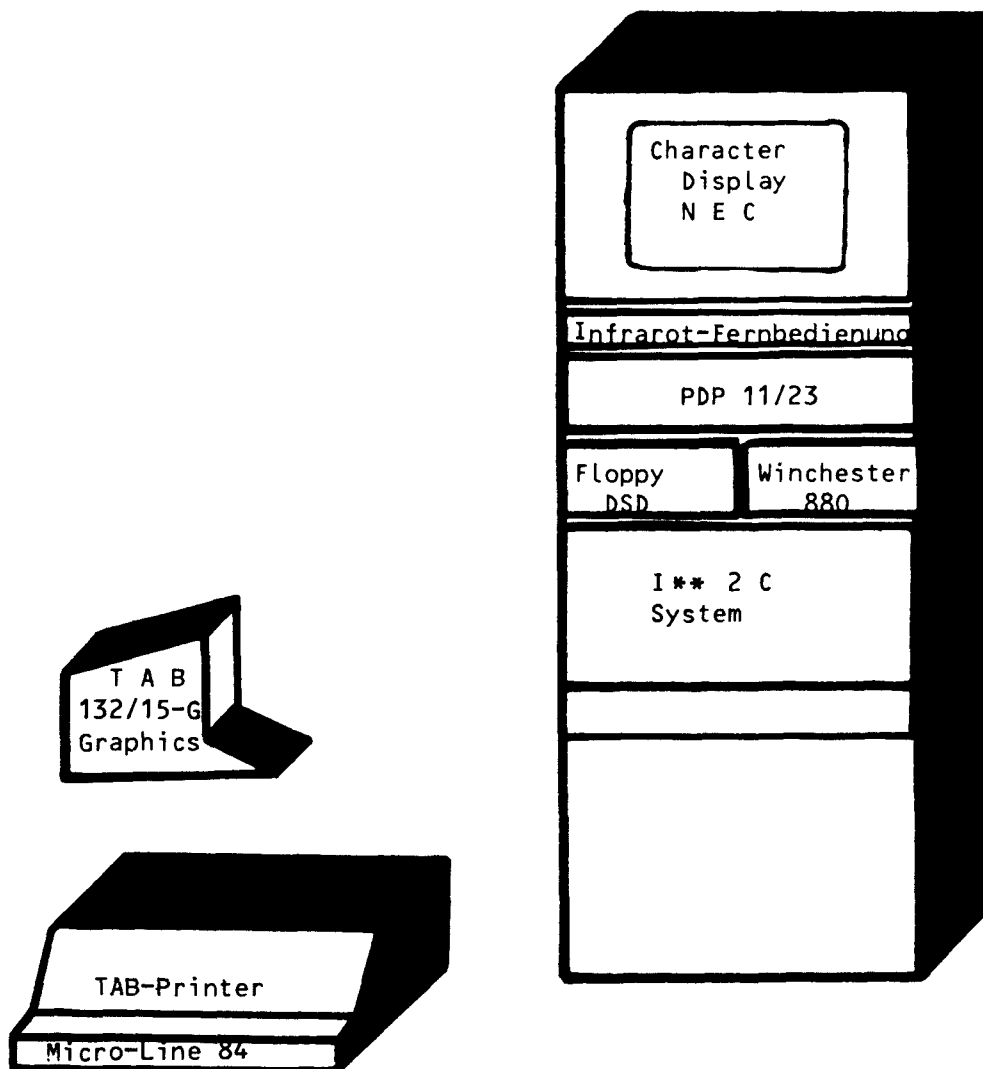
**Fig. 7: Blockschaltbild des ADC-DAC-Moduls**

Der DAC liefert an seinem Ausgang eine Spannung von 0-10V oder einen Strom von 4-20mA. Durch ein I<sup>2</sup>C-Bus-Kommando kann zwischen Spannungs- und Stromausgabe umgeschaltet werden. Zur Ansteuerung des X-t-Schreibers wird nur der 0-10V-Ausgang benutzt. Die auf dem X-t-Schreiber darzustellende Zählrate wird ebenfalls über ein I<sup>2</sup>C-Bus-Kommando in den DAC geschrieben.



## Kapitel 5

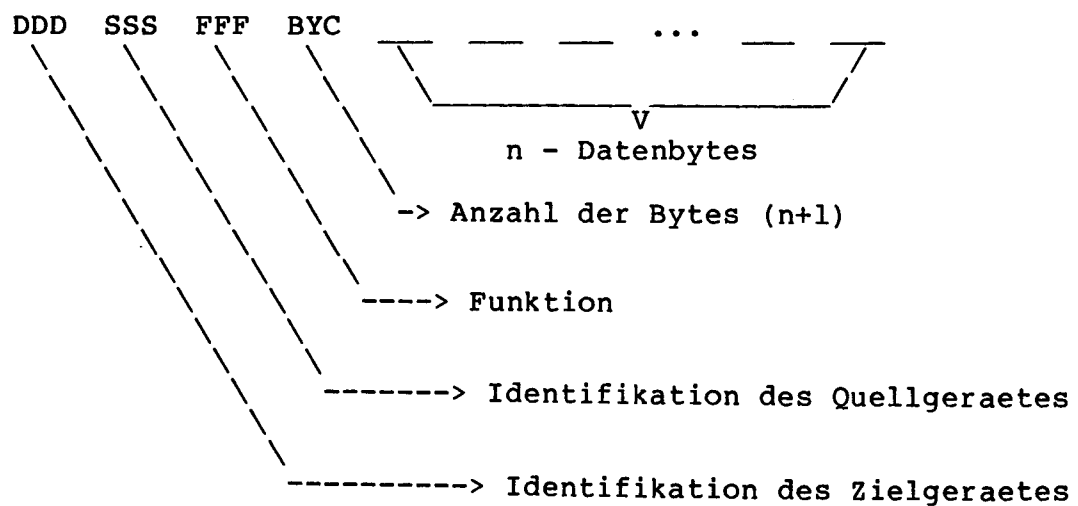
### DIE RECHNERKONFIGURATION



## Kapitel 6

### EXPERIMENT - RECHNER I\*\*2 C BUS

Das Protokoll zwischen den Anwenderprogrammen auf der Experimentseite und dem I\*\*2 C Bus hat folgende Struktur:



Source und Destination-Identifizierungen sind wie folgt spezifiziert:

204	Display Interface
220	Zaehler
230	Motor 1
232	Motor 2
234	Motor 3
236	Motor 4
240	Motor 5
250	ADC/DAC

Hardware-mässig stellt ein DL11 die Nahtstelle zwischen Experiment und I\*\*2 C Bus dar, so als handele es sich um eine normale Terminal Ein/Ausgabe. Die gewählte Baudrate von 2400 Baud gewährleistet eine sichere Synchronisation zwischen dem RSX11/M Betriebssystem und der 8085 Display Konvertierungsroutine, welche die ASCII Daten in binärer Form auf den seriellen Bus legt und in umgekehrter Richtung die seriellen binären Informationen des I\*\*2 C Bus in ASCII Formate dem RSX11/M Terminal Handler anbietet. Die folgenden Auszüge aus einigen Anwenderprogrammen, sollen diesen Formalismus verdeutlichen:

```

C      Blanko-String fuer Motorsteuerung
C      *****
DATA MOTS/'2','M','M',' ','2','0','4',' ','F','F','F',' ',
1      '0','0','B',' ','D','D','D',' ','D','D','D',' ',
2      '0','0','0'/

C      Motor-Destination Adressen
C      *****
DATA MORE/'3032','3436','4000'/

C      Schreibe Befehl auf seriellen Bus
C      *****
500    CONTINUE
      WRITE (1,1) (MOTS(M),M=1,IFOR)
1      FORMAT ('+'27A1)

C      Lies eine Antwort vom seriellen Bus mit Time-Out Behandlung
C      *****
CALL WTQIO (RLB, 1, 1,, ISTA, PARM)
IF (ISTA(1) .NE. 2) GO TO 33
INCO = INNO*10
WRITE (5,34) PARA(2), PARA(1), INCO
34    FORMAT(' Timeout bei Counter : '1I1,' Funktion : '1I2,' nach :
1      1I2,' Sekunden!')
IERR = ISTA(1)
RETURN

C      Anzahl Bytes :
C      *****
DATA BYLA/15,19,23,39, 19,23,31,0, 15,0,0,0, 15,0,0,0,
1      15,19,23,39, 23,31,0,0, 15,0,0,0, 15,0,0,0/

```

## Kapitel 7

### DAS BETRIEBSSYSTEM RSX11M 4.0

#### 7.1 DIE SYSTEMGENERIERUNG

Bei diesem Projekt wurde ein Betriebssystem RSX 11M in der Version 4.0 für Rechner aus der Familie PDP 11 von der Firma Digital Equipment Cooperation ( DEC ) generiert.

Die Generierung eines " mapped standard function systems " mit einem original RL01 Distribution Kit von DEC erfolgte auf einem Host Rechner ( PDP 11/45 ).

Die Betriebssystem Konfiguration umfasste dabei die Komponenten :

- Prozessortyp : PDP 11/23 mit 124 kW Memory
- Ohne Floating Point Prozessor ( FP-11 )
- Ohne extended instruction set
- Line Frequenz von 60 Hz
- Ohne KW11-Y watchdog support und memory parity support
- Höchster Interrupt Vektor : 774 ( oktal )
- Verschiedene Device Controller :
  1. 1 Controller für 2 RL01/02 Disk Laufwerke : DL
  2. 1 Controller für 2 RK03/05 Disk Laufwerke : DK
  3. 1 Controller mit DOUBLE density für 2 RX02 Floppy Disk Laufwerke : DY
  4. 3 Controller für ein DL11 Asynchronous Terminal Interface: YL
  5. 1 Konsol Device : Co



6. 1 Null Device : NL

Das Target-System hat einen Hintergrundspeicher vom Typ DSD 880, der aus einer DEC-kompatiblen RL02 Winchester-Platte und einem RX02 Floppy-Disk Laufwerk besteht.

Nach der Generierung auf dem Host-System wurde das bootable RSX11M-System mit Hilfe eines QBUS-UNIBUS Konverters von einer RL02 auf die Winchester-Platte des Target-Rechners transferiert.

7.2 STARTUP - AKTIVITÄTEN

Das Betriebssystem kann sowohl softwaremässig als auch hardwaremässig von der Winchester-Platte gebootet werden. Nach dem Bootvorgang meldet sich das RSX11M-System ( Startup-Commandfile ) mit der Frage nach der aktuellen Uhrzeit und dem Datum :

PLEASE ENTER TIME AND DATE ( hh:mm dd-mmm-yy ) : \_

Um die Real-Time Uhr des Rechners auf die aktuelle Zeit einzustellen, sollte man hier die Angaben machen. Der default-Wert sind die Daten des letzten gesicherten Systems.

## Kapitel 8

### SOFTWARE - ENTWICKLUNGEN

Nach den Anforderungen musste zum einen ein Software-paket entworfen werden, das die angeschlossene Hardware unterstützte und testete, zum anderen sollte für den späteren Anwender ein möglichst benutzerfreundliches Programmpaket angeboten werden. Das letztere verlangte vor allem ein Software-Interface für die Kommunikation mit den I\*\*2 C-Bus Modulen, das anwendungsbezogen aufrufbar ist.

Somit lassen sich diese Entwicklungen in zwei Bereiche trennen :

1. Der Entwicklerbereich
2. Der Anwenderbereich

Damit ergab sich eine logische Struktur mit verschiedenen Ebenen, die den Anwender ( obere Ebene ) von den betriebssystem- und hardware-internen Details befreit.

#### 8.1 ENTWICKLER SEITE

Für den Entwickler bildete das Software-Interface zur I\*\*2 C-Bus Peripherie den Schwerpunkt. Damit war es möglich, mit der Hardware-Seite zu kommunizieren. Testroutinen erlaubten die Überprüfung der angeschlossenen Hardware-Module. Für jeden Modultyp existieren so entsprechende Routinen, die das spezifische Handling übernehmen. Die Struktur der Entwicklerseite zeigt die umseitige Abbildung.

I\*\*2 C-Bus  
 \*\*\*\*\*

LSI 11/23  
 \*\*\*\*\*

Peripherie -----	Dispatcher -----	Modulspezifische Software -----
STEP-CO 1		Motorsteuerung M O T O
STEP-CO 2		Achsenposition kl. A X D A
STEP-CO 3	DISPLAY MODUL mit 8085 Processor  als Dispatcher	Achsenposition gr. L A D A
STEP-CO 4		
SCALER 1		Scalersteuerung C O U N
		Counterdarstellung C O D A
ADC/DAC 1		ADC/DAC-Steuerung A D A C

## 8.2 ANWENDER SEITE

Dieser Bereich beinhaltet die Routinen für das Mess- und Steuerprogramm für das Experimentier-Strahlrohr I am  $\gamma$ -Diffraktometer GADI.

Die Aufgaben des Programms lassen sich wie folgt zusammenfassen :

1. Vorbereitung zur Steuerung und Datenerfassung
  - \* Initialisieren der Motorpositionen
  - \* Anlegen eines Direct Access Datensatzes
2. Steuerung und Datenerfassung mit
  - \* Positionierung der Stepper-Motoren
  - \* Datenspeicherung
  - \* Datendarstellung

Die umseitige Abbildung zeigt die logische Struktur dieses Bereiches.

**Modulspezifische Routinen**  
 -----

**Anwenderbezogene Routinen**  
 -----

Motorsteuerung  
 M O T O

Mitteilungen fuer den User  
 M E S A

Achsenposition kl.  
 A X D A

Ja / Nein Beantwortung  
 E N T S

Achsenposition gr.  
 L A D A

Dialog Experimentparameter  
 D I A L

Scalersteuerung  
 C O U N

ASCII-INTEGER Umwandlung  
 A S I N

Counterdarstellung  
 C O D A

ADC/DAC-Steuerung  
 A D A C

A  
 N  
 W  
 E  
 N  
 D  
 E  
 R  
 P  
 R  
 O  
 G  
 R  
 A  
 M  
 M

## Kapitel 9

### SOFTWARE

#### 9.1 I\*\*2 C-BUS SOFTWARE

Die in Abschnitt 4 beschriebenen Mess- und Steuergeräte sind alle über ein I\*\*2 C-Bus-Interface an den I\*\*2 C-Bus angeschlossen. Dieser Bus ist ein serieller 2-Draht-Bus, über den Kommandos, Fertigmeldungen und Fehlermeldungen zwischen den am Bus angeschlossenen Modulen übertragen werden.

Das I\*\*2 C-Bus-Interface besteht aus dem 1-Chip-Microcomputer MAB 8400 der Firma Valvo. Die Software für diesen Microcomputer steht in einem EPROM (Erasable Programmable Read Only Memory) mit einem Speicherbereich von 2 Kilobyte.

Die Programme in den einzelnen Modulen haben alle den gleichen Aufbau. Sie bestehen im Wesentlichen aus folgenden Routinen:

1. Empfangsroutine: Diese empfängt Kommandos vom I\*\*2 C-Bus und speichert diese in einem Zwischenspeicher.
2. Senderoutine: Die Senderoutine sendet Antworten auf den I\*\*2 C-Bus. Diese Antworten werden vorher von anderen Routinen im Zwischenspeicher aufgesetzt.
3. Funktionsroutinen: Diese werden durch die verschiedenen Kommandos aufgerufen und bringen die Kommandos in der am Microcomputer angeschlossenen Hardware zur Ausführung. Ausserdem bereiten die Funktionsroutinen die Fertigmeldungen und eventuelle Daten (z.B. beim Auslesen eines Zählers) im Zwischenspeicher vor.
4. Hauptprogramm: Das Hauptprogramm erwartet den Empfang von Kommandos. Wird ein Kommando empfangen, wird dieses auf Fehler überprüft und falls kein Fehler festgestellt wird ruft das Hauptprogramm die entsprechende Funktionsroutine auf. Nach Ausführung der Funktion wird die Senderoutine aufgerufen und die Antwort auf den I\*\*2 C-Bus gesendet. Stellt das Hauptprogramm nach Empfang eines Kommandos einen Fehler fest, so bereitet es eine Fehlermeldung vor und ruft sofort die Senderoutine auf. Nach dem Aussenden der Antwort wartet das Hauptprogramm auf weitere Kommandos. Wird durch ein Kommando eine Funktion aufgerufen, die zur Ausführung längere Zeit beansprucht (z.B. Fahren eines Steppermotors), so löst die

angeschlossene Hardware nach Ablauf dieser Zeit (z.B. beim Erreichen der Sollposition) beim Microcomputer einen Interrupt (Unterbrechung) aus, wodurch eine zweite Antwort auf den I\*\*2 C-Bus gegeben wird.



## 9.2 MOTORSTEUERUNG

Die funktionsspezifische Bearbeitung zur Motorsteuerung übernimmt das Unterprogramm MOTO .

```

C      Aufruf :
C      SUBROUTINE MOTO (PARA, IERR)
C
C      Parameter :
C      PARA(2) > Motornummer
C      PARA(1) > Funktions-Kennzahl
C      PARA(3) > Anzahl der Schritte
C              > 1 Step = 10**-2 mm, max.: 25250 Steps
C      IERR    > Error-Kennung
C
C*****
C      Aufgabe : Motorsteuerung ueber 5 Steppercontroller
C              mit den Funktionen
C      Funktionsnummer    Funktionsbeschreibung
C      -----
C      00                > Gib Modulidentifikation
C      01                > emergency stop
C      02                > decelerating stop
C      03                > single step forward (clockwise)
C      04                > single step backward
C      05                > accelerating/decelerating mo. forward
C      06                > accelerating/decelerating mo. backward
C      07                > constant speed forward
C      08                > constant speed backward
C      09                > move until limit forward
C      10                > move until limit backward
C      11                > move until highspeed limit forward
C      12                > move until highspeed limit backward
C      13                > move until base point forward
C      14                > move until base point backward
C      15                > read finish data
C      16                > read input signal
C      17                > read output signal
C      18                > read remaining step number
C      19                > read encoder input
C      20                > reset stepper controller
C      21                > reset encoder input
C      22                > parallel load
C      23                > initialize st.co. with set of parameters
C*****
C      Deklarationen
C      -----
C      INTEGER *2 STEP, POAX(5), IFUN, IFOR, IZAH, PARA(8), RIUM(2),
C      1      NAXI, COAX, IERR, ISTA(2), PARM(6), RLB, IFAN
C      REAL *4      RCOU(8), ZAH, MORE(3), REIN
C      LOGICAL *1 OKAS(8), FUNC(3), MOTS(27), BYCO
C      LOGICAL *1 EMFE(40), MODE(2,5)
C      LOGICAL *1 ERMES(3,4), W, F, NULL
C
C      EQUIVALENCE (MOTS(17), OKAS), (BYCO, MOTS(15)),
C      1      (MOTS(9), FUNC), (MODE, MORE),

```

```

2          (RIEQ, RIUM)

COMMON /C1/ POAX, RCOU          ! Motorposition und Counterstand

DATA ASC1/11*'0'/, RLB/"1200/, NULL/'0'/
DATA ERMES/'D','e','S','E','n','d','F','n','s','O','f','f'/
1          , W/.TRUE./,F/.FALSE./

C          Blanko-String fuer Motorsteuerung
C          *****
DATA MOTS/'2','M','M',' ','2','0','4',' ','F','F','F',' ','
1          '0','0','B',' ','D','D','D',' ','D','D','D',' ','
2          '0','0','0'/

DATA MORE/'3032','3436','4000'/ ! Motor-Destination Adressen

COAX = 1
MOTS(1) = '2'
IFUN = PARA(1)
MONU = PARA(2)
CALL GETADR (PARM, EMFE)

IERR = 0          ! Anfangsbedingungen
IFAN = 0
IF (IFUN .NE. 23) GO TO 9

8          DO 8 I=1,3          ! Motoren auf Maximalgeschwindigkeit
MOTS(I) = NULL
.
.
.
C          Schreibe Befehl auf seriellen Bus
C          *****
500         CONTINUE
WRITE (1,1) (MOTS(M),M=1,IFOR)
1          FORMAT ('+'27A1)
.
.
.
C          Funktion 22(octal)
C          *****
700         ZAHL = 16.
CALL A S I N (EMFE, ZAHL)

IZAHL = INT(ZAHL) + 1
IF (MONU .NE. 1) GOTO 800
POAX(1) = 25253
IF (MERK .EQ. -1) POAX(1) = 0

CALL A X D A (MONU)
CALL L A D A (COAX, MONU)

800        RETURN
END

```

### 9.3 ZÄHLER ANSTEUERUNG

Das Unterprogramm COUN behandelt sämtliche Funktionen des Scaler - Moduls auf dem I\*\*2 C Bus :

```

C      *****
C      Aufruf :
C      SUBROUTINE C O U N ( PARA, IERR )
C      Funktionbezeichner      Funktions Nummer
C      -----
C      START COU                1
C      STOP COU                 2
C      READ COU                 3
C      READ TI-DAY              4
C      START TI-DA              5
C      SET TI-DAY               6
C      SET OUT-FRE              7
C      SET COU VAL              8
C      SET COU REL              9
C      GATE DISABL              10
C      GATE COU-CO              11
C      RES GROU CO              12
C      STA ZAEH RA              13
C      STO ZAEH RA              14
C
C      Parameter :
C      PARA(1) > Nummer der Funktion
C      PARA(n) > Feld fuer Daten
C      IERR    > Error-Kennung
C      *****
C      REAL *4 SEFE(8), RLAF(11), ZAHL, RCOU(8), CORE
C      LOGICAL *1 LSEF(32), LAFU(3,14), OKAS(8), DREI(3),
C      1      EMFE(40), ERBY(3), ASLO(3), ASHI(3), ASZA(6)
C      INTEGER *2 PARA(8), MELA(14), IFUN, IERR, LAFO, POAX(5),
C      1      ISTA(2), PARM(6), RLB, NCOU, INRE(2)
C      *****
C      Common fuer Motorposition und Counter
C      *****
C      COMMON /C1/ POAX, RCOU
C
C      EQUIVALENCE (SEFE, LSEF), (RLAF, LAFU),
C      1      (EMFE(9), ERBY),
C      2      (EMFE(17), ASLO), (EMFE(21), ASHI),
C      3      (INRE(1), CORE)
C
C      DATA SEFE/'220 ','204 ','FFF ','BYC ','DDD ','DDD ',
C      1      'DDD ','DDD '/
C      DATA RLAF/'0020','0200','2001','0010','0500','3004',
C      1      '0040','0200','3002','0010','0100'//, RLB/'1200/
C      DATA MELA/1,1,1,0,0,4,2,3,3,1,2,1,0,0/
C      DATA ERBY/'0','0','0'/
C      DATA DREI/'0','0','3'/

```

```

IERR = 0
IFUN = PARA(1)
INRE(1) = PARA(3)
INRE(2) = PARA(4)
C
C   Gueltige Funktion
C   *****
IF ((IFUN .GE. 1) .AND. (IFUN .LE. 14)) GOTO 11
IERR = - 1
RETURN
C
11  CALL GETADR (PARM, EMFE)
C
C   Umwandlung Zahl in ASCII-Darstellung und Lade Funktionsfeld
C   *****
REIN = PARA(1)
CALL R I A S (OKAS, REIN)
C
DO 10 I=1,3
10  LSEF(8+I) = OKAS(I)
    LSEF(10) = LSEF(10) + 4
C
C   Lade Byte-Count
C   *****
DO 20 I=1,3
20  LSEF(12+I) = LAFU(I, IFUN)
C
C   Funktions - Dispatcher
C   *****
GO TO (100, 100, 100, 200, 200, 300,
1    400, 500, 500, 100, 400, 100, 600, 600) IFUN
C
100  REIN = PARA(2)
    CALL R I A S (OKAS, REIN)
C
DO 30 I=1,3
30  LSEF(16+I) = OKAS(I)
GO TO 800
.
.
.
CALL WTQIO (RLB, 1, 1,, ISTA, PARM)
C
IF (ISTA(1) .NE. 2) GO TO 33
C
INCO = INNO*10
WRITE (5,34) PARA(2), PARA(1), INCO
34  FORMAT(' Timeout bei Counter : '111,' Funktion : '112,' nach : '
1    112,' Sekunden!')
IERR = ISTA(1)
RETURN
.
.
.
1000 RETURN
END

```

#### 9.4 ADC - DAC - ANSTEUERUNG

Die funktionsspezifische Bearbeitung für ein ADC/DAC - Modul übernimmt das Unterprogramm ADAC .

```

C      Aufruf :
        SUBROUTINE ADAC (PARA, IERR)

C      Parameter :
C          IFUN = PARA(1) > Funktions-Kennzahl
C          MODE = PARA(2) > Betriebsart
C          IDAT = PARA(3) > Parameter
C          IERR > Error-Kennung
C
C      Offset = 64
C      *****
C      Deklarationen
C      -----
        REAL *4 SEFE(10), ADCD, VABR(8)
        LOGICAL *1 EMFE(40), OKAS(8), VABY(4,8), LSEF(40), ERBY(3)
        INTEGER *2 PARA(8), IFUN, IERR, PARM(6), RLB, MODE,
1          BYLA(4,8), LAFO, ISTA(2)

        EQUIVALENCE (VABY, VABR), (LSEF, SEFE), (EMFE(9), ERBY)
C      Blanko-String fuer ADC-steuerung
C      *****
        DATA SEFE/'250 ','204 ','1FF ','00B ','DDD ','DDD ','DDD ',
1          'DDD ','DDD ','DDD '/
        DATA ERBY /'0', '0', '0'/
        DATA RLB/"1200/
C      Anzahl der Parameter :
        DATA VABR/'1237','235 ','1 ','1 ','1237',
1          '35 ','1 ','1 '/

        IERR = 0
        IFUN = PARA(1)
        MODE = PARA(2)

        IF ((IFUN .GE. 1) .AND. (IFUN .LE. 8)) GO TO 11
        IERR = -1
        RETURN

11      CALL GETADR (PARM, EMFE)
        REIN = IFUN
        CALL RIAS (OKAS, REIN)           ! Lade Funktionsfeld
        DO 10 I=1,3
10      LSEF(8+I) = OKAS(I)

        LSEF(9) = LSEF(9) + 1
        LSEF(15) = VABY(MODE, IFUN)     ! Lade Byte-count

C      Funktions-Dispatcher
C      *****
        GO TO (100, 200, 100, 100, 100, 100, 100, 100) IFUN

100      RETURN

```

```
200      IF (MODE .EQ. 2) GO TO 210
          REIN = PARA(3)
          CALL RIAS (OKAS, REIN)
          DO 220 I=1,3
220          LSEF(16+I) = OKAS(I)
          REIN = PARA(4)
          CALL RIAS (OKAS, REIN)

          DO 230 I=1,3
230          LSEF(20+I) = OKAS(I)
          REIN = PARA(5)
          CALL RIAS (OKAS, REIN)

          DO 240 I=1,7
240          LSEF(24+I) = OKAS(I)
          GO TO 1000

210      REIN = PARA(3)
          CALL RIAS (OKAS, REIN)

          DO 250 I=1,7
250          LSEF(16+I) = OKAS(I)

1000     LAFO = BYLA(MODE, IFUN)

          WRITE (1,1) (LSEF(I), I=1,LAFO)
1          FORMAT('+', 32A1)

          PARM(2) = 40
          INNO = 1
          PARM(3) = INNO
          CALL WTQIO (RLB, 1, 1,, ISTA, PARM)
          IF (ISTA(1) .NE. 2) GO TO 33

          INCO = INNO*10
          WRITE (5,34) PARA(1), INCO
34          FORMAT(' Timeout bei ADC Funktion : '1I2,' nach : '
1          1I2,' Sekunden!')
          IERR = ISTA(1)
          RETURN

33      DO 80 I=1,3
          IF (ERBY(I) .EQ. '0') GOTO 80
          IERR = IERR - 1
80      CONTINUE
          IF (IERR .EQ. 0) RETURN
          IERR = -1
          RETURN
          END
```

## 9.5 DISPLAY - ANSTEUERUNG

Neben den Unterprogrammen zur Kommunikation mit den I\*\*2 C-Bus-Modulen zur Steuerung und Datenerfassung existieren einige Routinen, die ein sogenanntes Display-Modul ansteuern. Dieses Modul stellt die Verbindung zu einem Bildschirm her, auf dem die momentanen Achsenpositionen der Steppermotoren und die Zählraten der angeschlossenen Counter dargestellt werden. Bei der Darstellungsform kann man zwischen Grossschrift und Normalschrift wählen. Je nach Schriftgrösse und Darstellungstyp werden verschiedene Unterprogramme aufgerufen.

### 9.5.1 Unterprogramm LADA

Die Aufgabe dieses Unterprogramms ist die Darstellung der momentanen Positionen der Achsen (1-5) und die aktuellen Zählerstände in einer Gross-Schrift auf dem Bildschirm.

```

C      Aufruf :
C      SUBROUTINE L A D A (COAX, GENU)

C      Parameter :
C      COAX > Kennung fuer Counter(2), Axis(1)
C      GENU > Index der Achse (1,2,3,4,5) bzw. der Counter (1..8)

C      Common :
C      C1 / Poax(n), Rcou(m)

C      Steuerzeichen fuer Display :
C      2 = ^B > Gross-Schrift
C      14 = ^N > Cursor-Positionierung
C      9 = ^I > Haelftenumschaltung

C      Deklarationen
C      -----
C      INTEGER *2 GENU, POAX(5), COAX
C      LOGICAL *1 POSTG(20), CHAR(4), BLA, PUNK, SEZE
C      REAL *4 HIST, RCOU(8)

C      Common fuer Motorposition und Counter
C      *****
C      COMMON /C1/ POAX, RCOU

C      DATA POSTG/0,9,2,14,39,' ','0','0','0',' ','+', '?','?','?',' ','1',
C      1 ' ','?','?','?',' ','3,9/
C      DATA CHAR/'A','X','C','O'//, BLA/' '/, SEZE/' '//
C      DATA PUNK/'.'/

C      IF ( (COAX .EQ. 1) .AND. (GENU .GT. 5) ) RETURN
C      IF ( (COAX .EQ. 2) .AND. (GENU .GT. 8) ) RETURN
C      IF (COAX .EQ. 2) COAX = COAX+1
C      POSTG(7) = CHAR(COAX)
C      POSTG(8) = CHAR(COAX+1)
C      POSTG(9) = GENU + 48
C      POSTG(15) = PUNK
C      POSTG(18) = BLA

```



```
      IF (GENU .EQ. 4) POSTG(18) = SEZE

      IF (COAX .GT. 1) GO TO 100  ! Berechne aktuellen Wert
C      *****
C      M o t o r e n
C      *****
      POSTG(12) = BLA
      IHIL = IABS(POAX(GENU))
      J = 18
      DO 20 I=1,5
      IF ((I .EQ. 3) .AND. (GENU .NE. 4)) J = 17
      POSTG(J-I) = MOD(IHIL,10) + 48
20     IHIL = IHIL/10
      POSTG(11) = '+'
      IF (POAX(GENU) .LT. 0) POSTG(11) = '-'
      GO TO 14

C      *****
C      Z a e h l e r
C      *****
100    HIST = RCOU(GENU)
      J = 16
C      POSTG(J-1) = AMOD(HIST,10.) + 48
C      IHIL = HIST/10.
      DO 30 I=1,5
      POSTG(J-I) = AMOD(HIST,10.) + 48
30     HIST = HIST/10

      POSTG(16) = BLA
      POSTG(17) = BLA

14     WRITE (1,4) POSTG
4       FORMAT ('+',20A1)

      RETURN
      END
```

```
                                HR:  MIN:  SEC:
C01  #7:00016    #2:
AX1:+000.00    AX2:+000.00    AX3:+
AX4:+ 00480"    AX5:+

      AX4:+ 00480"

10000*
1000*
100*
```

Der Monitor stellt einige Experimentparameter dar.  
Zählerinhalte und Achsenpositionen

```
#204 220 001 003 013 000 043
#204 220 001 003 015 000 045
#204 220 001 003 011 000 041
#204 220 001 003 014 000 044
#204 220 001 003 015 000 045
```

Kontrollausgabe der ASCII-Schnittstelle auf dem Bildschirm  
204 => Destination / 220 => Source / 1 => Funktion / usw.

### 9.5.2 Unterprogramm CODA

Während das obengenannte Unterprogramm LADA die Darstellung in Gross-Schrift durchführt, gibt die Routine CODA die momentanen Zählerstände in Normalschrift auf den Display.

```
C      Aufruf :
      SUBROUTINE C O D A (NCOU)

C      Parameter :
C      NCOU > Nummer des Counter (1 bis 8)

C      Common :
C      C1 / Poax(n), Count(m)

C      Steuerzeichen fuer Display :
C      9 = ^I > Umschaltung des Displays
C      14 = ^N > Cursor-Positionierung
C
C      Deklarationen
C      -----
      INTEGER *2 NCOU, POAX(5)
      REAL *4 RCOU(2), HIST
      LOGICAL *1 POST(13)

C
C      Common fuer Motorposition und Counter
C      *****
      COMMON /C1/ POAX, RCOU

      DATA POST/0,9,14,'$',"46",' ',':',',','?',',','?',',','?',',','?',',','?',9/

      POST(6) = NCOU + 48
C      Zeige den aktuellen Counter-Wert
C      *****
      HIST = RCOU(NCOU)
      J = 13
      DO 20 I=1,5
      POST(J-I) = AMOD(HIST,10.) + 48
20      HIST = HIST/10.

      WRITE (1,4) POST
4      FORMAT ('+',13A1)

      RETURN
      END
```

### 9.5.3 Unterprogramm BACO

Diese Routine stellt den momentanen Zählerstand als Balkendiagramm auf einem Display dar.

```

C      Aufruf :
C      SUBROUTINE BACO (NCOU)
C      Parameter :
C      NCOU > Nummer des COUnters (1 bis 8)
C      COMMON : C1 / POAX(5), RCOU(8)

C      Steuerzeichen fuer den Display :
C      9 = ^I > Haelftenumschaltung
C      14 = ^N > Cursorpositionierung
C      Deklarationen
C      -----
C      INTEGER *2 NCOU, POAX(5)
C      LOGICAL *1 MIN, DEL, BLA
C      LOGICAL *1 BALZ(22), BALT(20), BALH(21)
C      LOGICAL *1 BLEZ(10), BLET(10), BLEH(10)
C      REAL *4 RCOU(8)
C      COMMON fuer Motorpositionen und Counter
C      *****
C      COMMON /C1/ POAX, RCOU
C      EQUIVALENCE (BALZ(13), BLEZ)
C      EQUIVALENCE (BALT(11), BLET)
C      EQUIVALENCE (BALH(11), BLEH)

C      DATA BALZ/0,9,14,'1',' ','1','0','0','0','0','*',',',' ',10*',' '/
C      DATA BALT/ 14,'2',' ',' ','1','0','0','0','*',',',' ',10*',' '/
C      DATA BALH/ 14,'3',' ',' ','1','0','0','0','*',',',' ',10*',' ',9/

C      DATA BLA/' '/, DEL/127/

C      IF ((NCOU .GT. 8) .OR. (NCOU .LT. 1)) RETURN
C      DO 10 I=1,10
C      BLEZ(I) = BLA
C      BLET(I) = BLA
10    BLEH(I) = BLA
C      IZIF = IDIV/10
C      .
C      .
C      IF (IZIF .LT. 1) GO TO 41
C      DO 40 I=1,IZIF
40    BLEH(I) = DEL

41    WRITE (1,6) BALZ
C      WRITE (1,5) BALT
C      WRITE (1,4) BALH
4    FORMAT ('+',21A1)
5    FORMAT ('+',20A1)
6    FORMAT ('+',22A1)
C      RETURN
C      END

```

#### 9.5.4 Unterprogramm AXDA

Analog zum obengenannten Unterprogramm CODA stellt die Routine AXDA die momentanen Achsenpositionen ( 1 <= Achse <= 5 ) in Normalschrift dar.

```

C      Aufruf :
C      SUBROUTINE A X D A (NAXI)
C      Parameter :
C      NAXI > Index der Achse (1,2,3,4,5)
C      Common :
C      C1 / Poax(n), Count(m)
C      Steuerzeichen fuer Display :
C      *****
C      9 = ^I > Umschaltung des Displays
C      14 = ^N > Cursor-Positionierung
C      & > Zeile fuer Ax1 bis Ax3
C      ( > Zeile fuer Ax4 und Ax5
C      2 = 62 > Spalte fuer Ax2, Ax5
C      $ = 36 > Spalte fuer Ax1, Ax4
C      A = 65 > Spalte fuer Ax3
C      Deklarationen
C      INTEGER *2 NAXI, POAX(5), HIST
C      REAL *4 RCOU(8)
C      LOGICAL *1 POST(14), POSP(5), POZE(2), SEZE, BLAN
C      Common fuer Motorposition und Counter
C      *****
C      COMMON /C1/ POAX, RCOU
C      DATA POST/0,9,14,'Y','X','+', '?', '?', '?', '.', '?', '?', ' ',9/
C      DATA POSP/'$', '2', 'A', '$', '2'/
C      DATA POZE/'&', '('/, SEZE/'"/, BLAN/' '/

C      POST(13) = BLAN
C      POST(10) = '.'
C      POST( 7) = BLAN
C      IF (NAXI .EQ. 4) POST(13) = SEZE
C      IF (NAXI .LE. 3) POST(4) = POZE(1)
C      IF (NAXI .GT. 3) POST(4) = POZE(2)
C      POST(5) = POSP(NAXI)
C      Show current Motor-Position
C      *****
C      HIST = IABS(POAX(NAXI))
C      J = 13
C      DO 20 I=1,5
C      IF ((I .EQ. 3) .AND. (NAXI .NE. 4)) J = 12
C      POST(J-I) = MOD(HIST,10) + 48
20    HIST = HIST/10
C      POST(6) = '+' ! Vorzeichen der Position
C      IF (POAX(NAXI) .LT. 0) POST(6) = '-'
C      WRITE (1,4) POST
4    FORMAT ('+',14A1)
C      RETURN
C      END

```

## 9.6 ALLGEMEINE UNTERPROGRAMME

### 9.6.1 Dialog mit dem Anwender über die Experimentparameter

Das Unterprogramm DIAL behandelt den Dialog mit dem Anwender für die Eingabe der aktuellen Experimentparameter.

```
C      Aufruf :
      CALL D I A L (LAUS)

C      Parameter :
C      LAUS  > Kennung fuer Entscheidung zur Aenderung der Parameter

C      COMMON : COPA

C      Unterprogramm - Aufruf :
C      ADJU  > String rechtsbuendig schreiben
C
C      Deklarationen
C      -----
      INTEGER *4 FELA, RECS, MESZEI, PRIZEI, XTME, PRIWER, ROMO
      REAL      *4 BLOX, BLOY, HORI, VERT, ENST, TEMP, HODE, VEDE, GIPA,
1          ROTI, RODE, MATE, MARO, MAHO, MAVE, MAGI, MABX, MABY
      LOGICAL *1 KOMM(80), DATU(9), ZEIT(8), EXOR(20), ARTQ(6), PRMA(6)
1          PRIK, NMES(6), REFL(3), PRON(20), INKR(4),
2          W, F, DIDA(500), LDUM(35), BLA, LENT, LAUS
      CHARACTER *4 CPRI
C      *****
C      C O M M O N - Bereich
C      *****
      COMMON/COPA/ KOMM, DATU, ZEIT, NMES, EXOR, ARTQ,
1          PRON, PRMA, REFL, INKR, PRIK, LDUM,
2          ENST, BLOX, BLOY, TEMP, ROMO, ROTI,
3          RODE, HORI, HODE, VERT, VEDE, GIPA,
4          MESZEI, PRIZEI, XTME, PRIWER

      DATA BLA/ ' ' /, W/ .TRUE./, F/ .FALSE./, EXOR/ 20* ' ' /, REFL/ 3* '
1 , NMES/ 6* ' ' /, PRON/ 20* ' ' /, ARTQ/ 'I','r', 4* ' ' /
2 , PRMA/ 6* ' ' /

C      Input Conversation Error Handling
C      *****
      CALL ERRSET (64, W, F, W, F)

C
C      Maximale Grenzen fuer Parameter
C      *****
      MABX = 5.0          ! Blendengroessen
      MABY = 20.0
      MATE = 2000.        ! Temperatur
      MARO = 144000.      ! Rotationsachse
      MAHO = 253.         ! Horizontalachse
      MAVE = 253.         ! Vertikalachse
      MAGI = 20.          ! Gitterparameter
```

```
      NUMP = 0

      WRITE (5,1)
1      FORMAT (/' Eingabe von Parametern und Kenndaten ',
1          /1X,35('*')/)

      IF (LAUS) GO TO 155

52     WRITE (5,2)
2     FORMAT ('$Gib Datum      (dd-mmm-yy)           : ')
      READ (5, 3, END=1000, ERR=52) IQ, DATU
3     FORMAT (Q,9A1)
      IF (IQ .EQ. 0) CALL DATE (DATU)
      IF (NUMP .GT. 0) GO TO 155

54     WRITE (5,4)
4     FORMAT ('$Gib Uhrzeit      (hh:mm:ss)           : ')
      READ (5, 5, END=1000, ERR=54) IQ, ZEIT
5     FORMAT (Q, 8A1)
      IF (IQ .EQ. 0) CALL TIME (ZEIT)
      IF (NUMP .GT. 0) GO TO 155

56     WRITE (5,6)
6     FORMAT ('$Gib Nummer der Messung (Nr.Version) : ')
      READ (5, 7, END=1000, ERR=56) IQ, NMES
7     FORMAT (Q, 6A1)
      IF (IQ .EQ. 0) NMES(1) = '1'
      FELA = 6
      CALL A D J U (NMES, FELA)
      IF (NUMP .GT. 0) GO TO 155
      .
      .
      .
      .
      .
      .
      .
      .
      .
544    WRITE (5, 44)
44     FORMAT('$Rotationsmotor (Fein = 4 v Grob = 3) : ')
      READ (5, 45, END=1000, ERR=544) IQ, ROMO
45     FORMAT (Q, 11)
      IF (IQ .EQ. 0) ROMO = 4
      IF (NUMP .GT. 0) GO TO 155

546    WRITE (5, 46)
46     FORMAT('$Xt-Schreiberbereich 10**n. Gib n : ')
      READ (5, 47, END=1000, ERR=546) IQ, XTME
47     FORMAT (Q, 11)
      IF (IQ .EQ. 0) XTME = 4
      IF ((XTME .LT. 1) .OR. (XTME .GT. 5)) GO TO 546
      IF (NUMP .GT. 0) GO TO 155
```



```
C      *****
C      Kontroll - Ausgabe
C      *****
149    WRITE (5,150)
150    FORMAT (//5X,'Kontroll - Ausgabe der Parameter'/1X,41('*')
1      '/' Par.Num.',1X,'Par.Bez.',16X,'Par.Wert')

      I5 = 1
      I4 = 2
151    WRITE (5, 151) I5, DATU, I4, ZEIT, NMES, EXOR, ARTQ
      FORMAT (I7,3X,'Datum',18X,9A1/I7,3X,'Zeit',20X,8A1/
1      6X,'3',3X,'Messung',19X,6A1/6X,'4',3X,'Benutzer',4X,20A1
2      /6X,'5', 3X,'Quelle', 20X, 6A1)

      WRITE (5,152) PRON, PRMA, ENST, BLOX, BLOY, REFL, TEMP
152    FORMAT(6X,'6',3X,'Nr.Probe',4X,20A1/6X,'7',3X,'Material',18X,6A1
1/6X,'8',3X,'Energie',15X,F10.3/6X,'9',3X,'Blenden Breite',8X
1,F10.3
1 /5X,'10',3X,'Blenden Hoehe',9X,F10.3
2 /5X,'11',3X,'Reflex',23X,3A1/5X,'12',3X,'Temperatur',
3 11X,F11.3)
      WRITE (5,153) ROTI, RODE, HORI, HODE, VERT, VEDE, INKR, GIPA, MES
153    FORMAT(5X,'13',3X,'Rota.Seku.',12X,F10.3/5X,'14',3X,'Delta
1 Rota.',10X,F10.3/
2 5X,'15',3X,'Hor.Trans.',12X,F10.3/5X,'16',3X,'Delta
3 Trans.',9X,F10.3/5X,'17',3X,'Ver.Trans.',12X,F10.3/
5 5X,'18',3X,'Delta Trans.',9X,F10.3/5X,'19',3X,
6 'Kont. od. Inkr.',13X,4A1/5X,'20',3X,'Gitterpar.',16X,
7 F6.3/5X,'21',3X,'Messzeit (sek)',13X,I5)
      WRITE (5,154) CPRI, PRIZEI, ROMO, XTME
154    FORMAT (5X, '22', 3X, 'Primaerstrahlkennung '7X, A4
1/5X,'23',3X,'Messzeit Primaerstrahl',5X,I5
2/5X,'24',3X,'Rotation (Fein=4, Grob=3)',5X,I2
3/5X,'25',3X,'Xt-Messbereich (10**n)',5X,I5//)

155    WRITE (5,156)
156    FORMAT ('$Gib Nummer fuer zu aendernden Parameter : ')
      READ (5,157, END=1000, ERR=155) IQ, NUMP
157    FORMAT (Q, I2)
      IF ((IQ.EQ. 0) .OR. (NUMP.EQ. 0)) GO TO 149
      IF ((NUMP.LT. 0) .OR. (NUMP.GT. 25)) RETURN

      GO TO (52,54,56,58,510,512,514,516,518,540,520,522,5240,5260,524
1, 526, 528, 530, 532, 534, 536, 538, 542, 544, 546) NUMP

1000   RETURN
      END
```

### 9.6.2 Ja / Nein Beantwortung

Das Unterprogramm **ENTS** setzt eine logischen Variable wahr oder falsch je nachdem, ob eine Frage mit Ja oder Nein beantwortet wurde.

```

C      Aufruf :
C      CALL E N T S (LENT)

C      Parameter :
C      Name          Erklaerung          Typ      In-Output
C      ----          -
C      LENT      > True (Ja) oder False (nein)      L*1      (Output)

C      Deklarationen
C      -----
C      LOGICAL *1 LENT, T, F
C      CHARACTER *4 CENT, NEIN, JA, knein, kja

C      DATA NEIN/ 'NEIN'//, JA/ 'JA '/, knein/ 'nein'//, kja/ 'ja '/
C      DATA T/.TRUE./, F/.FALSE./

C      CALL ERRSET (64, T, F, T, F)

51     WRITE (5, 1)
1      FORMAT ('$  Gib JA oder NEIN : ')
      READ (5, 2, ERR=51, END=1000) IQ, CENT(1:IQ)
2      FORMAT (Q, A)
      IF ( (IQ .EQ. 0) .OR. (IQ .GT. 4) ) GO TO 51

      IF ( (CENT(1:) .EQ. JA(1:)) .OR. (CENT(1:) .EQ. kja(1:)) )
1      THEN
          LENT = T
          RETURN
      ELSEIF ((CENT(1:).EQ.NEIN(1:)) .OR.
1      (CENT(1:) .EQ. knein(1:))) THEN
          LENT = F
          RETURN
      ENDIF
      GO TO 51
1000   LENT = F
      RETURN
      END

```

### 9.6.3 Initialisierung des Counters im Scaler-Modul

Das Unterprogramm INCO initialisiert den Counter 7 und 8 eines Scaler-Moduls mit der Busadresse 220 in den Timer-Mode.

```
C      Aufruf :
C      CALL I N C O (MEZI, IERR)

C      Parameter :
C      MEZI > Messzeit
C      ierr > Fehlerkennung

C      Deklarationen
C      -----
C      INTEGER *2 MEZI, ierr, PARA(8)
C      REAL *4 CORE

C      EQUIVALENCE (PARA(3), CORE)
C      EQUIVALENCE (PARA(1), IFUN)
C      EQUIVALENCE (PARA(2), MOCO)

C      Initialisiere Counter 7 und 8 mi Messzeit MEZI
C      *****
C      IERR = 0
C      IFUN = 13
C      MOCO = 3
C      CORE = MEZI*100.
C      PARA(8) = MEZI
C      CALL C O U N ( PARA, IERR )
C      MENU = 3
C      IF (IERR .NE. 0) CALL M E S A (MENU, MOCO)
C      IFUN = 14
C      MOCO = 1
C      CALL C O U N ( PARA, IERR )
C      IF (IERR .NE. 0) CALL M E S A (MENU, MOCO)

C      RETURN
C      END
```

#### 9.6.4 Mitteilungen an den Anwender

Das Unterprogramm MESA druckt eine Nachricht auf das Anwenderterminal.

```
SUBROUTINE MESA (MENU, MOCO)
C      Parameter :
C      MENU  >  Message Nummer
C      MOCO  >  Motor oder Counter-Nummer

      GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) MENU
1      WRITE (5, 70) MOCO, IERR
70     FORMAT (' Einschalten des Counters',I5,' Fehler:',I5)
      RETURN
2      WRITE (5, 80) MOCO, IERR
80     FORMAT (' Lesen des Counters',I5,' Fehler:',I5)
      RETURN
3      WRITE (5, 90) MOCO, IERR
90     FORMAT (' Reset des Counters',I5,' Fehler:',I5)
      RETURN
4      WRITE (5, 110) MOCO, IERR
110    FORMAT (' Fahre zum Endschalter Motor',I5,' Fehler:',I5)
      RETURN
5      WRITE (5, 120) MOCO, IERR
120    FORMAT (' Fahren des Motors',I5,' Fehler:',I5)
      RETURN
6      WRITE (5, 130)
130    FORMAT (' Nicht alle Module online !')
      RETURN
7      WRITE (5,100)
100    FORMAT(///' '36('*')/' Gamma - Diffraktometer Messprogramm '
1/' '36('*')//)
      RETURN
8      WRITE (5,140)
140    FORMAT (' Default DIAL-Datei auslesen ?')
      RETURN
9      WRITE (5,150)
150    FORMAT (' Endschalter wird angefahren !/' '29('*'))
      RETURN
10     WRITE (5,160)
160    FORMAT (' Handjustierung beendet ? ')
      RETURN
11     WRITE (5, 170)
170    FORMAT (' Soll Messdaten-File angelegt werden ?')
      RETURN
12     WRITE (5, 180)
180    FORMAT (' Fehler bei der Motorinitialisierung.')
      RETURN
13     WRITE (5, 190)
190    FORMAT (' Xt-Schreiber-Messbereich zu klein. '/34('*'))
      RETURN
14     WRITE (5,200)
200    FORMAT(' Detektor in Primaerstrahlposition ?')
      RETURN
15     CONTINUE
      RETURN
```

#### 9.6.5 Aufnahme von Zählraten

Das Unterprogramm MESS nimmt über eine bestimmte Messzeit Zählraten aus einem Scaler Modul des I\*\*2 C Bus auf.

```
C      Aufruf :
C      CALL M E S S (MIZE, WERT, IERR)

C      Parameter :
C      MIZE  >  MEZE - Messzeit
C      WERT  >  REAL Zaehlrates
C      IERR  >  Fehlerkennung

C      Deklarationen
C      -----
      INTEGER *2 MIZE, IERR, PARA(8), PARM(6), RLB, ISTA(2)
      REAL *4 WERT, CORE
      LOGICAL *1 EMFE(40)

      EQUIVALENCE (PARA(3), CORE)
      DATA RLB /"1200/

      CALL GETADR (PARM, EMFE)

C      Zaehler 8 im Timer-Mode
C      *****
      PARA(1) = 13
      PARA(2) = 1
      CORE = MIZE*100.          ! => PARA(3)
      CALL C O U N ( PARA, IERR)
      IF (IERR .NE. 0) GO TO 1020

      PARM(2) = 40
      PARM(3) = MIZE/10 + 1

      CALL W T Q I O ( RLB, 1, 1,, ISTA, PARM)
      IF (ISTA(1) .EQ. 2) GO TO 1020

      ZAHL = 16.
      CALL ASIN (EMFE, ZAHL)

      PARA(1) = 14
      PARA(2) = 1
      CALL C O U N ( PARA, IERR )
      IF (IERR .NE. 0) GO TO 1020
      WERT = ZAHL
      RETURN

1020      INCO = ( MIZE/10 + 1 ) * 10
      WRITE (5,34) PARA(2), PARA(1), INCO
34      FORMAT(' Timeout bei Counter : ', I1, ' Funktion : ', I2,
1          ' nach : ', I4, ' Sekunden!')

      RETURN
      END
```

## Kapitel 10

### DER MESSABLAUF

Bevor ein Kristall mit dem Gamma-Diffraktometer untersucht werden kann, muss durch eine röntgenographische Voruntersuchung die Netzebenenorientierung bis auf ca.  $\pm 2$  Grad bekannt sein.

#### 10.1 DIE HANDJUSTIERUNG

Der Kristall wird nach dieser Vororientierung so auf das Goniometer positioniert, dass nur noch kleine Rotationsbewegungen notwendig sind, um den gewünschten BRAGG-Reflex beobachten zu können.

Der Detektor wird nun von Hand um den Winkel  $2\theta_B$  (entsprechend  $\lambda = 2d_{hkl} \sin \theta_B$ ) ausgelenkt. Mit der Fernbedienung wählt man dann eine der drei möglichen Motorengeschwindigkeiten vor und dreht den Kristall mit der Grobrotation. Bei der Suche des Reflexes wird eine grosse Kollimatorspaltbreite benutzt, um auch schwache Reflexe zu finden. Durch gleichzeitiges Beobachten eines Ratemeters, das die im Detektor registrierten Impulse anzeigt, erkennt man optisch und akustisch, wann der Kristall in Reflexposition steht.

Ist diese gefunden, geht man zur Feinrotation über und dreht den Kristall nach beiden Seiten aus dem Reflex heraus bis sichergestellt ist, dass nur noch Untergrundimpulse in den Detektor gelangen. Beide Positionen werden an einer Mikrometerschraube abgelesen und aus der Differenz erhält man den Winkelbereich, über den die Rockingkurve gemessen werden soll. Der Kristall wird dann mit der Fernbedienung in die Anfangsposition gefahren.

#### 10.2 DIE AUFNAHME VON ROCKINGKURVEN

Welcher Kollimatorspalt bei der Messung benutzt wird, richtet sich nach den Intensitätsverhältnissen bei der Reflexsuche und wird von Fall zu Fall durch die Erfahrung des Experimentators bestimmt. Erwartet man eine breite Rockingkurve, so werden grössere Spalte benutzt, bei "guten" Kristallen wählt man entsprechend schmalere Spalte, um mit einer guten Strahldivergenz (minimal  $10''$ ) den Reflex auszumessen und eventuell vorhandene Nebenpeaks zu erkennen. In Abhängigkeit vom benutzten Spalt wird dann die Schrittweite  $\Delta\omega$  (Vielfaches von  $2''$ ) gewählt. Die Messzeit pro Punkt bestimmt dann die statistische Sicherheit der einzelnen Zählraten.

Handelt es sich um einen grösseren Kristall, der an verschiedenen Stellen untersucht werden soll, so gibt man die gewünschten Positionen in den Dialog des Messprogramms ein.

```
Kontroll - Ausgabe der Parameter
*****
Par.Num.  Par.Bez.                      Par.Wert
  1      Datum                      12-JUL-84
  2      Zeit                        17:19:48
  3      Messung                      1
  4      Benutzer                     GAIN
  5      Quelle                       IR
  6      Nr.Probe                     NI-840614
  7      Material                     NI
  8      Energie                      468.000
  9      Blenden Breite               1.000
 10      Blenden Hoehe               15.000
 11      Reflex                       220
 12      Temperatur                   293.000
 13      Rota.Seku.                   16200.000
 14      Delta Rota.                  40.000
 15      Hor.Trans.                   20.000
 16      Delta Trans.                 20.000
 17      Ver.Trans.                   0.000
 18      Delta Trans.                 40.000
 19      Kont. od. Inkr.              Inkr
 20      Gitterpar.                   3.524
 21      Messzeit (sek)               10
 22      Primaerstrahlkennung         1
 23      Messzeit Primaerstrahl       5
 24      Rotation (Fein=0, Grob=1)    0
 25      Xt-Messbereich (10**n)       4
```

Bedienungsdialog für eine Messung

Oft werden Kristalle bis zu ihrem endgültigen Einsatz den verschiedensten Behandlungsformen unterworfen. (Ätzen, Biegen, Glühen, Beladung mit Wasserstoff,...). Es ist dann von Interesse zu wissen, wie sich die einzelnen Vorgänge auf die Rockingkurven, also auf die Mosaikverteilung auswirken. Durch eine Markierung am Kristall ist es dann später möglich jede beliebige Stelle exakt anzufahren und mit den ursprünglichen Kurven zu vergleichen.

Für bestimmte Messungen (Vergleich mit theoretischen Modellen, Vergleich unterschiedlicher Kristalle des gleichen Materials,...) benötigt man das sogenannte integrale Reflexionsvermögen. Das ist im Wesentlichen die auf den ungebeugten Strahl bezogene Fläche unter der Rockingkurve. Dazu wird die Probe in die Startposition d.h. weit ausserhalb des Bragg-Reflexes gedreht und der Detektor von Hand in Position Null (Primärstrahlposition) gebracht. Hier wird 5 mal die Primärstrahlintensität mit installierter Probe gemessen. Bei diesem Verfahren ist also die Kenntnis des Absorptionsfaktors nicht notwendig. Der Rechner speichert dann den Mittelwert der Primärstrahlungsmessung und fragt, ob die Handjustierung beendet ist. Dazu stellt man den Detektor wieder in Reflexposition, gibt ein "Ja" ins Terminal und hat die Möglichkeit den Datensatz mit einer Kommentarzeile zu versehen :

Gib JA oder NEIN : JA

Gib Kommentarzeile (max. 79 Zeichen) :

1.....10.....20.....30.....40.....50.....60.....70.  
1. MESSUNG 5 CM VON DER FLACHEN KRISTALLFLÄCHE

Nach Abschicken des Kommentars beginnt der Rechner mit der automatischen Messung an den vorgegebenen Stellen. Während der Messung erscheinen laufend die aktuellen Werte der Positionen, der Zeit und des Detektors auf dem Bildschirm und können bei Bedarf auf dem angeschlossenen Drucker protokolliert werden. Der DA-Wandler erlaubt ausserdem den Verlauf der Rockingkurven auf einem X/t-Schreiber zu verfolgen.

Nach dem letzten Messpunkt laufen alle Motoren in ihre Ausgangsposition zurück und ein neuer Kristall kann untersucht werden.



### 10.3 DER INITIALISIERUNGS-DIALOG

Im Wesentlichen läuft das Unterprogramm DIAL, in dem alle benötigten Experiment Parameter interaktiv eingelesen werden. Um nicht alle Parameter eingeben zu müssen, kann ein Default Parameter-Datensatz eingelesen werden :

Es erscheint auf dem Bildschirm die Frage :

SOLL DEFAULT PARAMETERDATEI GELESEN WERDEN ?

GIB JA ODER NEIN : \_\_\_\_\_

Wird an dieser Stelle " JA " eingegeben, erscheint nach dem Einlesen der Default-Datei eine Kontrollausgabe der default Werte am Bildschirm. Der Anwender können aktuelle Werte verändert werden. Diese aktuellen Experiment Parameter werden in dem ersten Satz des Messdatensatzes abgelegt.

\*\*\*\*\*  
Gamma - Diffraktometer Messprogramm  
\*\*\*\*\*

Default DIAL-Datei auslesen ?  
Gib JA oder NEIN : JA  
Gib Nummer fuer zu aendernden Parameter :

Kontroll - Ausgabe der Parameter  
\*\*\*\*\*  
Par.Num. Par.Bez. Par.Wert  
1 Datum 12-JUL-84  
2 Zeit 17:19:48  
3 Messung 1  
4 Benutzer GAIN  
5 Quelle IR  
6 Nr.Probe NI-840614  
7 Material NI  
8 Energie 468.000  
9 Blenden Breite 1.000  
10 Blenden Hoehe 15.000  
11 Reflex 220  
12 Temperatur 293.000  
13 Rota.Seku. 16200.000  
14 Delta Rota. 40.000  
15 Hor.Trans. 20.000  
16 Delta Trans. 20.000  
17 Ver.Trans. 0.000  
18 Delta Trans. 40.000  
19 Kont. od. Inkr. Inkr  
20 Gitterpar. 3.524  
21 Messzeit (sek) 10  
22 Primaerstrahlkennung 1  
23 Messzeit Primaerstrahl 5  
24 Rotation (Fein=0, Grob=1) 0  
25 Xt-Messbereich (10\*\*n) 4

Gib Nummer fuer zu aendernden Parameter : 13  
Rotations-Gesamtschritte (Seku) : 16000.  
Gib Nummer fuer zu aendernden Parameter : 26  
Gib Messdaten-Dateinamen (FILE.EXT;VER) : 840712.DAT;1  
Detektor in Primaerstrahlposition ?  
Gib JA oder NEIN : JA  
Handjustierung beendet ?  
Gib JA oder NEIN : JA  
Gib Kommentarzeile (max. 79 Zeichen) :  
1.....10.....20.....30.....40.....50.....60.....70.....79  
1. MESSUNG 5 CM VON DER FLACHEN KRISTALLFLAECH

Beispiel eines Dialoges

Bei der späteren Auswertung muss deshalb zunächst der erste Satz des Direct-Access-Files gelesen werden, um alle Informationen über die Aufteilung der weiteren Sätze zu erhalten. Vom Beginn der Messung wird fortlaufend ein Messprotokoll auf der Konsole ausgegeben. Der Experimentator kann den Verlauf der Messung verfolgen und notfalls abbrechen. Abb. 6 zeigt einen solchen Kontrollabschnitt.

```
Gib Messdaten-Dateinamen (FILE.EXT;VER) : 840718.DAT;1
Gib Kommentarzeile (max. 79 Zeichen) :
.....10.....20.....30.....40.....50.....60.....70.....79
1. MESSUNG 5 MM VOM FLACHEN KRISTALLENDE
Rotations-Messungen bei : X-Pos. : 0.00 Y-Pos. : 0.00
*****
Rotation : Z-Pos. : 0. Sekunden um 13:49:47 Uhr Counter : 6.
Rotation : Z-Pos. : 20. Sekunden um 13:49:59 Uhr Counter : 15.
Rotation : Z-Pos. : 40. Sekunden um 13:50:12 Uhr Counter : 10.
Rotation : Z-Pos. : 60. Sekunden um 13:50:24 Uhr Counter : 7.
Rotation : Z-Pos. : 80. Sekunden um 13:50:37 Uhr Counter : 8.
Rotation : Z-Pos. : 100. Sekunden um 13:50:49 Uhr Counter : 16.
Rotation : Z-Pos. : 120. Sekunden um 13:51:02 Uhr Counter : 8.
Rotation : Z-Pos. : 140. Sekunden um 13:51:14 Uhr Counter : 14.
Rotation : Z-Pos. : 160. Sekunden um 13:51:27 Uhr Counter : 11.
Rotation : Z-Pos. : 180. Sekunden um 13:51:39 Uhr Counter : 7.
Rotation : Z-Pos. : 200. Sekunden um 13:51:52 Uhr Counter : 13.
Rotation : Z-Pos. : 220. Sekunden um 13:52:05 Uhr Counter : 18.
Rotation : Z-Pos. : 240. Sekunden um 13:52:17 Uhr Counter : 9.
Rotation : Z-Pos. : 260. Sekunden um 13:52:30 Uhr Counter : 7.
Rotation : Z-Pos. : 280. Sekunden um 13:52:42 Uhr Counter : 14.
Rotation : Z-Pos. : 300. Sekunden um 13:52:55 Uhr Counter : 14.
Rotation : Z-Pos. : 320. Sekunden um 13:53:07 Uhr Counter : 12.
Rotation : Z-Pos. : 340. Sekunden um 13:53:20 Uhr Counter : 7.
```

Protokoll einer Messung

## Kapitel 11

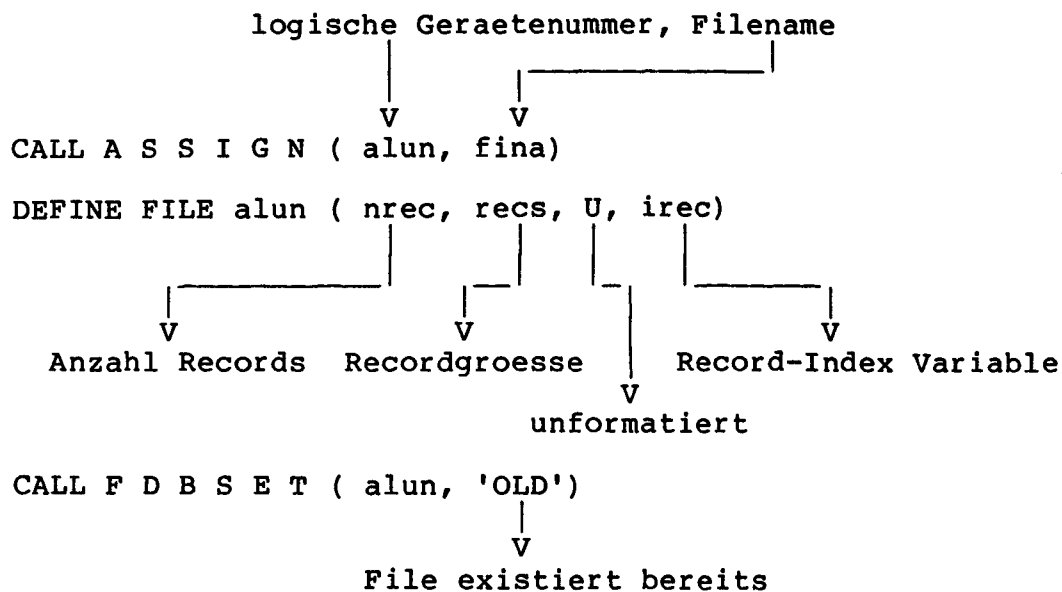
### DIREKT ACCESS FILES

#### 11.1 ZWECK

Die Messdaten und die jeweiligen Experimentparameter werden in Messdateien gespeichert, die als Direct-Access-Files (DA-File) organisiert sind. Diese Organisationsform wurde gewählt, um später schnell auf gezielte Messdaten zuzugreifen.

#### 11.2 DEKLARATION

Die Eröffnung eines existierenden DA-File's in FORTRAN zeigt die folgende Statement-Sequenz:



Die feste Satzlänge (Recordsize RECS) von 250 Worten (500 Bytes) ergab sich aus der Anforderung, bis zu 248 Messpunkte pro Einzelmessung innerhalb eines Satzes (Record) abzuspeichern. Zuzüglich der Messpunktkennungen ergibt sich damit die implementierte Satzlänge.

### 11.3 DATENFORMAT

Der DA-File besteht aus einzelnen Sätzen ( Record ). Ausser dem ersten Record, der die Experimentparameter enthält, werden innerhalb der Records die Messdaten und die Kennungen für die einzelnen Messpunkte abgelegt.

Der Aufbau des ersten Records zeigt folgendes Schema :

1. Record:

E X P E R I M E N T - P A R A M E T E R
---

Abgelegt wird hier der COMMON-Bereich COPA  
aus dem Hauptprogramm GADI

Der Aufbau des zweiten Records zeigt folgendes Schema :

2. Record:

INVE	INHO	ANRO	KENN	Date1	Date2	...	DateN	INVE...
2	2	2	2	4	4	...	4	2 Bytes

Ganzzahlige Daten [ INTEGER ] :

INVE := Index der vertikalen Position  
INHO := Index der horizontalen Position  
ANRO := Anzahl der Messdaten pro Messpunkt  
KENN := Kennung des letzten Messpunktes

Gleitkomma Daten [ REAL ] :

Date i := Am Messpunkt i erfasster Zaehlwert  
Die folgenden Records lassen sich nicht mehr einheitlich darstellen,  
da bei den fixen Satzlängen Daten und Kennungen pro Messpunkt in  
verschiedenen Records liegen können.

#### 11.4 BEARBEITUNG DES DA-FILES

Mit den Informationen der Experimentparametern kann man den existierenden DA-File lesen :

$$\text{Anzahl Rotationsmessungen} = 1 + \frac{\text{Rotationsweg}}{\text{Delta Rotationsweg}}$$

( ANRO )

$$\text{Anzahl Horizontalpositionen} = 1 + \frac{\text{Horizontalweg}}{\text{Delta Horizontalweg}}$$

( ANHO )

$$\text{Anzahl Vertikalpositionen} = 1 + \frac{\text{Vertikalweg}}{\text{Delta Vertikalweg}}$$

( ANVE )

Ein Block pro Messpunkt nimmt folgenden Speicherplatz ein :

$$\text{BLOC} = ( \text{ANRO} + 2 ) * 2 \quad [ \text{WORTE} = \text{Bytes} * 2 ]$$

Es ergibt sich ein Gesamtspeicherplatz für alle Messpunkte :

$$\text{GesamtWORTE} = \text{BLOC} * \text{ANHO} * \text{ANVE}$$

Ein bestimmter Datenrecord hat demnach den Recordindex von :

$$\text{IREC}_i = 2 + \frac{\text{BLOC} * I_{\text{Horizontal}} * I_{\text{Vertikal}}}{\text{RECS}}$$

{ RECS = 2\*250 [WORTE] }

$$\text{IREC}_{\text{Letzte}} = 2 + \frac{\text{Gesamt\_WORTE}}{\text{RECS}}$$

Das Programm LIDA liest einen vollständigen DA-File, in Verbindung mit dem Unterprogramm BUFO werden die Messdaten ausgegeben.

## Kapitel 12

### DIE AUSWERTUNG DER MESSDATEN

Die übersichtliche Darstellung der Zählraten, sowie die automatische Bestimmung der Halbwertsbreite, des integralen Reflexionsvermögens und des Verhältnisses Maximum zum Untergrund geschieht mit dem Programm LIDA.

Nach Angabe der Datenfiles werden alle wichtigen physikalischen Grössen berechnet und ausgegeben.

```
Kontroll - Ausgabe der Parameter
*****
Par.Num.  Par.Bez.                Par.Wert
   1      Datum                  12-JUL-84
   2      Zeit                   17:19:48
   3      Messung                 1
   4      Benutzer                GAIN
   5      Quelle                 IR
   6      Nr.Probe               NI-840614
   7      Material               NI
   8      Energie                468.000
   9      Blenden Breite         1.000
  10      Blenden Hoehe         15.000
  11      Reflex                 220
  12      Temperatur             293.000
  13      Rota.Seku.             16200.000
  14      Delta Rota.            40.000
  15      Hor.Trans.             20.000
  16      Delta Trans.           20.000
  17      Ver.Trans.             0.000
  18      Delta Trans.           40.000
  19      Kont. od. Inkr.        Inkr
  20      Gitterpar.             3.524
  21      Messzeit (sek)         10
  22      Primaerstrahlkennung    1
  23      Messzeit Primaerstrahl  5
  24      Rotation (Fein=0, Grob=1) 0
  25      Xt-Messbereich (10**n)  4
```

### Liste einer Parametergestaltung

Für eine eventuelle spätere Verwendung der Daten werden die Datenfiles auf Disketten übertragen und stehen so jederzeit zur Verfügung. Möchte man die Kurven graphisch darstellen, so besteht die Möglichkeit im IFF ( Gruppe Dr.Durcansky ) die Disketten einzulesen und die Daten zu einem Rechner mit Plot-Möglichkeit zu übertragen. Ein Beispiel zeigt die Fig. 8.

### CRYSTAL #403 (7) INFLUENCE OF ROTATION CHANGE

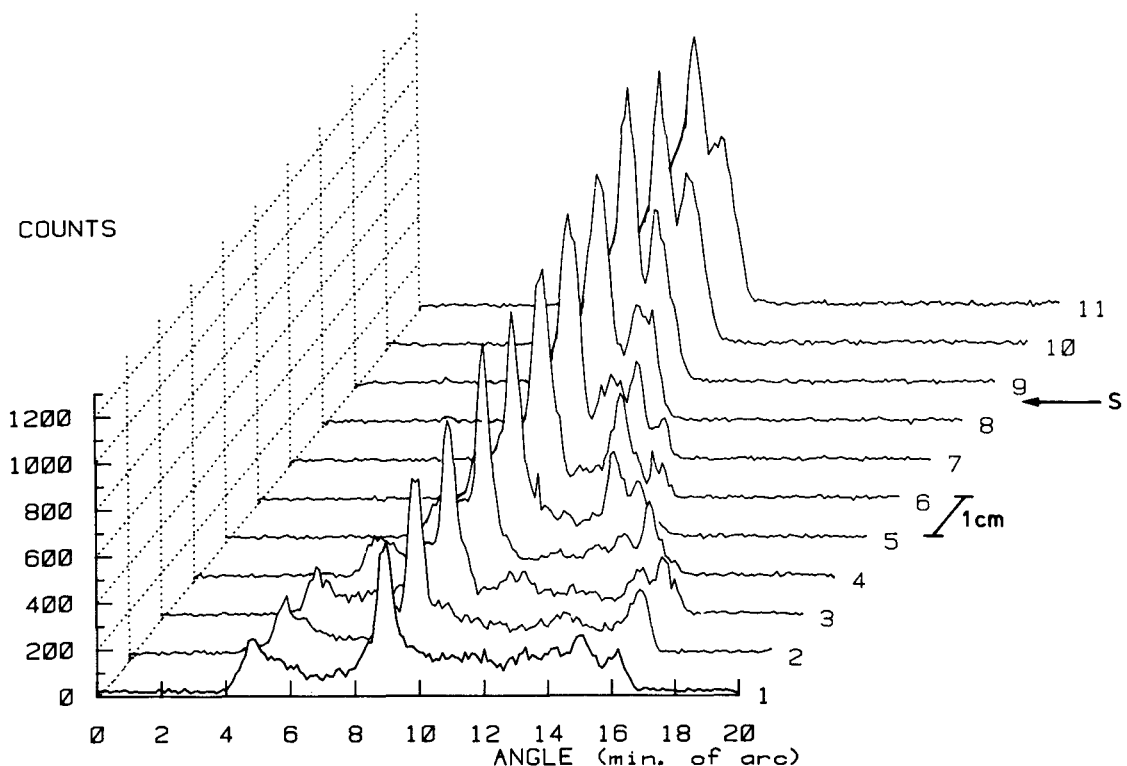


Fig. 8: Beugungsgebirge eines Cu<sub>3</sub>Au-Kristalls

Die Abbildung zeigt die Beugung der  $\gamma$ -Strahlen an den (220) - Netzebenen eines Cu<sub>3</sub>Au - Kristalls und lässt die Mosaikblockstruktur des Kristallgitters in Form eines Beugungsgebirges erkennen. Aufgetragen ist die gebeugte Intensität über dem Drehwinkel. Jeder Kurvenzug charakterisiert einen scheibenförmigen Bereich des zylinderförmigen Kristalls. Bei der durch Pfeil S gekennzeichneten Kristalllänge wurde während der Züchtung aus der Schmelze mit dem CZOCHRALSKI-Verfahren die Rotationsgeschwindigkeit geändert. Man erkennt deutlich eine drastische Verbreiterung des Beugungsgebirges. Auf diese Weise ist es mit dem  $\gamma$ -Diffraktometer möglich, zerstörungsfrei den Einfluss von Wachstumsparametern bei der Einkristallherstellung zu prüfen.

Die Daten, sowie die graphische Darstellung wurden freundlicherweise von Herrn Dr. W. Uelhoff, IFF, zur Verfügung gestellt.

```

)RUN LIDA
Gib Messdaten-Dateinamen (FILE.EXT;VER) : 840426.DAT;5
PRIK : 1 PRIR : 40221.00 PRIZ : 5
Kommentar :
PS = 65 GRAD
MEZE: 10 BLOX: 1.50
X-Pos. : 0.00 Y-Pos : 0.00
  0.      27.    1
 400.     18.    2
 800.     16.    3
1200.     24.    4
1600.     27.    5
2000.     21.    6
2400.     18.    7
2800.     68.    8
3200.    244.    9
3600.    649.   10
4000.   1014.   11
4400.    930.   12
4800.    828.   13
5200.    652.   14
5600.    574.   15
6000.    474.   16
6400.    319.   17
6800.    182.   18
7200.     92.   19
7600.     56.   20
8000.     29.   21
8400.     21.   22
8800.     30.   23
9200.     29.   24
9600.     18.   25
10000.    24.   26
10400.    28.   27
10800.    24.   28
11200.    17.   29
11600.    18.   30
12000.    28.   31

Summe der Zaehlraten: 6.50E+03

Untergrund: 20.67
Zaehlrate im Maximum: 1014.00
Verhaeltnis Maximum zu Untergrund: 4.91E+01

*****
Halbwertsbreite: 2400."
*****

Integrales Reflektionsvermoegen: 1.41E-04

```

### Beispiel 1: Teilauswertung einer Messung



200.	51.	3
300.	45.	4
400.	38.	5
500.	71.	6
600.	61.	7
700.	49.	8
800.	61.	9
900.	49.	10
1000.	59.	11
1100.	52.	12
1200.	69.	13
1300.	62.	14
1400.	62.	15
1500.	59.	16
1600.	61.	17
1700.	51.	18
1800.	74.	19
1900.	72.	20
2000.	76.	21
2100.	76.	22
2200.	86.	23
2300.	82.	24
2400.	89.	25
2500.	90.	26
2600.	96.	27
2700.	93.	28
2800.	98.	29
2900.	100.	30
3000.	88.	31
3100.	104.	32
3200.	113.	33
3300.	135.	34
3400.	116.	35
3500.	115.	36
3600.	125.	37
3700.	125.	38
3800.	118.	39
3900.	104.	40
4000.	108.	41
4100.	98.	42
4200.	92.	43
4300.	75.	44
4400.	87.	45
4500.	69.	46
4600.	63.	47
4700.	73.	48
4800.	81.	49
4900.	87.	50
5000.	81.	51
5100.	77.	52
5200.	86.	53
5300.	71.	54
5400.	80.	55
5500.	71.	56
5600.	63.	57
5700.	80.	58
5800.	81.	59
5900.	78.	60
6000.	76.	61
6100.	77.	62
6200.	85.	63
6300.	89.	64
6400.	67.	65
6500.	72.	66
6600.	67.	67
6700.	64.	68
6800.	50.	69
6900.	60.	70
7000.	49.	71
7100.	45.	72
7200.	65.	73
7300.	43.	74
7400.	36.	75
7500.	58.	76
7600.	39.	77
7700.	51.	78
7800.	33.	79
7900.	35.	80
8000.	34.	81

```

summe der Zaehlraten:      5.99E+03
Untergrund:                45.17
Zaehlraten im Maximum:    135.00
Verhaeltnis Maximum zu Untergrund:  2.99E+00
*****
albwertsbreite: 1600."
*****
Integrales Reflektionsvermoegen:  4.29E-05

```

```

) RUN LIDA
Gib Messdaten-Dateinamen (FILE.EXT:VER) : 840612.DAT;2
PRIZ : 1 PRIR : 26351.40 PRIZ : 10
Kommentar :
MITTE KRISTALL
MEZE: 10 BLDX: 1.00
X-Pos. : 0.00 Y-Pos : 0.00

```

0.	42.	1
100.	49.	2
200.	46.	3
300.	38.	4
400.	42.	5
500.	28.	6
600.	47.	7
700.	40.	8
800.	40.	9
900.	42.	10
1000.	34.	11
1100.	37.	12
1200.	51.	13
1300.	45.	14
1400.	40.	15
1500.	47.	16
1600.	38.	17
1700.	40.	18
1800.	61.	19
1900.	44.	20
2000.	52.	21
2100.	60.	22
2200.	54.	23
2300.	81.	24
2400.	70.	25
2500.	68.	26
2600.	102.	27
2700.	107.	28
2800.	98.	29
2900.	87.	30
3000.	97.	31
3100.	82.	32
3200.	92.	33
3300.	81.	34
3400.	95.	35
3500.	92.	36
3600.	85.	37
3700.	84.	38
3800.	88.	39
3900.	63.	40
4000.	64.	41
4100.	87.	42
4200.	77.	43
4300.	71.	44
4400.	77.	45
4500.	83.	46
4600.	95.	47
4700.	84.	48
4800.	90.	49
4900.	91.	50
5000.	100.	51
5100.	75.	52
5200.	72.	53
5300.	76.	54
5400.	68.	55
5500.	75.	56
5600.	65.	57
5700.	55.	58
5800.	65.	59
5900.	71.	60
6000.	66.	61
6100.	55.	62
6200.	53.	63
6300.	84.	64
6400.	65.	65
6500.	73.	66
6600.	87.	67
6700.	62.	68
6800.	70.	69
6900.	68.	70
7000.	68.	71
7100.	74.	72
7200.	65.	73
7300.	65.	74
7400.	50.	75
7500.	59.	76
7600.	66.	77
7700.	64.	78
7800.	69.	79
7900.	76.	80
8000.	56.	81

```

Summe der Zaehlraten:      5.43E+03
Untergrund:                56.33
Zaehlraten im Maximum:    107.00
Verhaeltnis Maximum zu Untergrund:  1.90E+00

```

## Beispiel 2: Teilauswertung einer Messung

## Kapitel 13

### TESTSOFTWARE FÜR I\*\*2 C BUS MODULE

Zu jedem Modultyp existiert ein Testprogramm, mit dem man die Funktionsausführung des entsprechenden Moduls überprüfen kann.

#### 13.1 MOTORTEST ( TEMO )

Mit diesem Programm lassen sich alle Funktionen eines beliebigen Stepper-Controllers auf dem I\*\*2 C-Bus testen. Von hier wird das funktionspezifische Unterprogramm MOTO aufgerufen, das die Kommunikation mit dem entsprechenden Modul übernimmt.

Die Steuerung des Testprogramms TEMO wird von einem Dialog übernommen, in dem verschiedene Parameter eingegeben werden :

- Die Nummer des zum Stepper-Controller gehörenden Stepper-motors.
- Die Nummer des gewünschten Funktioncodes für den Stepper-motor.
- Funktionsabhängige ( siehe Funktion 23 ) Parameter (1, 2, 3).
- Funktionsabhängige ( bei Funktion 5, 6, 7, 8 ) Wegstrecke.

Der jeweiligen Funktion entsprechend werden einige Ausgabedaten gelistet. Mit der Eingabe von CNTL Z anstelle des Parameters lässt sich das Programm abbrechen.

Die implementierten Funktionen sind aus der Tabelle der gültigen Funktioncodes (siehe 9.2. MOTO) zu entnehmen.

### 13.2 ZÄHLERTEST ( COTE )

Analog zum Testprogramm zur Motorsteuerung lassen sich auch bei diesem Programm alle Funktionen eines Scaler-Moduls auf dem I\*\*2 C-Bus überprüfen. Der Dialog behandelt die Eingabe verschiedener Parameter zum Testablauf :

- Die Nummer des Funktionscodes
- Funktionsabhängig :
  - \* Die Nummer des zu testenden Zählers
  - \* Die Zeit in dem Format hh:mm:ss
  - \* Die Zählfrequenz
  - \* Ein Zählerwert zur Initialisierung
  - \* Nummer eines Gates und Gruppe
  - \* Eine Zählrate in Sekunden

Bei entsprechenden Funktionen ( Zählerwerte ) werden die Rückgabedaten durch die Unterprogramme LADA und BACO auf einem Display dargestellt.

13.2.1 ADC / DAC - Modultest ( ADCT )

Es lassen sich mit dem Programm ADCT sämtliche Funktionen eines ADC/DAC - Moduls auf dem I\*\*2 C-Bus testen. Die Funktionsvielfalt dieses Moduls erfordert einen umfassenden Dialog mit dem Anwender.

- Die Nummer des Funktionscodes ( 0 bis 8 )
- Funktionsabhängig :
  - \* Die Nummer des gewünschten Parameter Modes (1 bis 5 ).
  - \* Je nach Mode die Eingabe verschiedener Parameter.

Die folgende Abbildung zeigt die Funktionstabelle mit der Anzahl von Parameter bei den verschiedenen Funktionsmodes :

Funktionen		M O D E S			
Beschreibung	Nr.	1	2	3	4
IDENT	0	0	-	-	-
READ	1	0	1	2	6
WRITE	2	1	2	4	-
RS IN MUX	3	0	-	-	-
RS OU MUX	4	0	-	-	-
RD SUB OFF	5	0	1	2	6
SET IN OUT	6	2	4	-	-
SHOW STATU	7	0	-	-	-
SELF TEST	8	0	-	-	-

## Kapitel 14

### DIE FILE-ORGANISATION

Die Datensätze für das Messprogramm befinden sich zusammen mit den zum Programm gehörigen Teilen auf einer 20MB Winchester Platte. Der Anwender muss sich unter der UIC [3,2] in das System einloggen und kann das Messprogramm dort starten. Der notwendige Datensatz mit den Experimentparametern ( DIAL.DAT ) muss sich unter dieser UIC befinden. Ein Messdatensatz wird von dem Messprogramm auch dort abgelegt.

Zur Sicherung von Datensätzen kann man über das zur Konfiguration gehörige Diskettenlaufwerk Datensätze auf Floppy-Disketten kopieren. Diese Disketten sollten aber nur als Zwischendatenträger dienen, da erfahrungsgemäss diese Disketten keine zuverlässigen Massenspeicher sind. Alle wichtigen Aufzeichnungen kann man von den Disketten auf Massenspeicher anderer Rechnern überspielen.

#### 14.1 DIE WINCHESTER PLATTE

Dieses Speichermedium enthält u.a. alle lauffähigen Programme und die dazugehörigen Quellprogramme. Zur besseren Übersicht sind die Arbeitsprogramme und die Quellprogramme unter verschiedenen UIC's abgelegt :

UIC [3,1] Directory : Software Tools

Directory DLO:[3,1]

LBR.TSK;1  
HPCAS.MAC;1  
TAB.MAC;1  
CIPTKB.CMD;1  
CIP11M.MAC;1  
CIP11M.OBJ;1  
TABT.CMD;1  
TAB.CMD;1  
CIP11M.CMD;1  
HPCAS.OBJ;1  
CIP11M.TSK;1

UIC [3,2] Directory : Arbeits - Account

Directory DLO:[3,2]  
12-JUL-84 17:33

DIAL.DAT;1	2.		18-MAY-83 15:05
LIDA.FTN;1	10.		22-MAY-84 10:52
840523.DAT;1	4.		23-MAY-84 08:40
MAIN.CMD;1	1.		15-JUL-83 10:39
SUBR.CMD;1	1.		15-JUL-83 10:39
SCRA.CMD;2	1.		15-JUL-83 10:39
INDI.FTN;1	7.		15-JUL-83 10:38
INDI.TSK;1	80.	C	10-MAY-83 15:42
MESA.FTN;1	4.		15-JUL-83 10:38
COUN.FTN;1	9.		15-JUL-83 10:38
MOTO.FTN;1	13.		18-JUL-83 16:00
DIAL.FTN;1	22.		15-JUL-83 10:38
BUFO.FTN;1	10.		06-JAN-83 17:57
AUSW.FTN;1	3.		06-JAN-83 17:57
GADI.OLB;1	159.	C	22-MAY-84 09:54
GADI.FTN;1	21.		22-MAY-84 09:55
GADI.LST;3	32.		22-MAY-84 09:55
GADI.TSK;3	113.	C	22-MAY-84 09:55
MESS.;1	0.	L	22-MAY-84 10:02
TEST.DAT;1	0.	L	15-JUN-84 16:48
TEST.DAT;2	4.		22-MAY-84 10:43
840523.DAT;2	4.		23-MAY-84 08:49
LIDA.LST;1	15.		22-MAY-84 10:53
840523.DAT;3	4.		23-MAY-84 08:54
LIDA.TSK;2	82.	C	22-MAY-84 10:54
840523.DAT;4	4.		23-MAY-84 09:37
840523.DAT;5	4.		23-MAY-84 09:45
840523.DAT;6	0.	L	23-MAY-84 11:09
840523.DAT;7	4.		23-MAY-84 11:10
840523.DAT;10	4.		23-MAY-84 11:23

UIC [3,3] Directory : Kommando Prozeduren

Directory DLO:[3,3]

MAIN.CMD;1  
SUBR.CMD;1  
SCRA.CMD;2  
GADI.OLB;1  
GADI.FTN;1  
GADI.LST;3  
GADI.TSK;3

UIC [3,4] Directory : Quell - Programme

Directory DLO:[3,4]

HOMO.FTN;1  
OKAS.FTN;1  
FILE.FTN;1  
TEEN.FTN;1  
ERTE.FTN;1  
MUSI.FTN;1  
ADJU.FTN;1  
TIME.FTN;1  
TIMO.FTN;1  
DITE.FTN;1  
KONT.FTN;1  
ADCT.FTN;1  
INTE.FTN;1  
INIT.FTN;1  
ENTS.FTN;1  
KOPF.FTN;1  
TEKO.FTN;1  
KOME.FTN;1  
PITE.FTN;1  
RIAS.FTN;1  
MESA.FTN;1  
ASIN.FTN;1  
TEMO.FTN;1  
BACO.FTN;1  
LADA.FTN;1  
AXDA.FTN;1  
MOTO.FTN;1  
TEFI.FTN;1  
CODA.FTN;1  
COTE.FTN;1  
COUN.FTN;1  
ADAC.FTN;1  
CREF.FTN;1  
INDI.FTN;1  
GADI.FTN;1

UIC [3,5] Directory : Lauffähige Programme (Testsoftware)

>PIP [3,5]\*.\*;\*/BR

Directory DLO:[3,5]

INDI.TSK;1  
LIDA.TSK;1  
GADI.TSK;1  
DIAL.DAT;1

## Kapitel 15

### AUSBLICK

Die weitere Planung sieht einen Ausbau der erfolgreichen Rechnersteuerung auch für den zweiten Messplatz am  $\gamma$ -Diffraktometer vor. Dort wird zur Zeit eine neue Kristallzuchtapparatur aufgebaut, um das Kristallwachstum während der Züchtung zu charakterisieren. Hierzu ist es notwendig, zusätzliche Parameter wie Druck, Temperatur, Temperaturgradienten usw. zu erfassen und gegebenenfalls zu steuern.

Die gemessenen Kurven sollen graphisch auf den Terminals dargestellt und mit Spline-Funktionen angepasst werden.

Bei Rockingkurven mit einer Überlagerung von mehreren Peaks ist es manchmal wünschenswert, die Einzelpeaks herauszupräparieren. Hierzu sollen Entfaltungsprogramme am KFA-Grossrechner zum Einsatz kommen.



Kapitel 16

FOTOGRAPHIEN

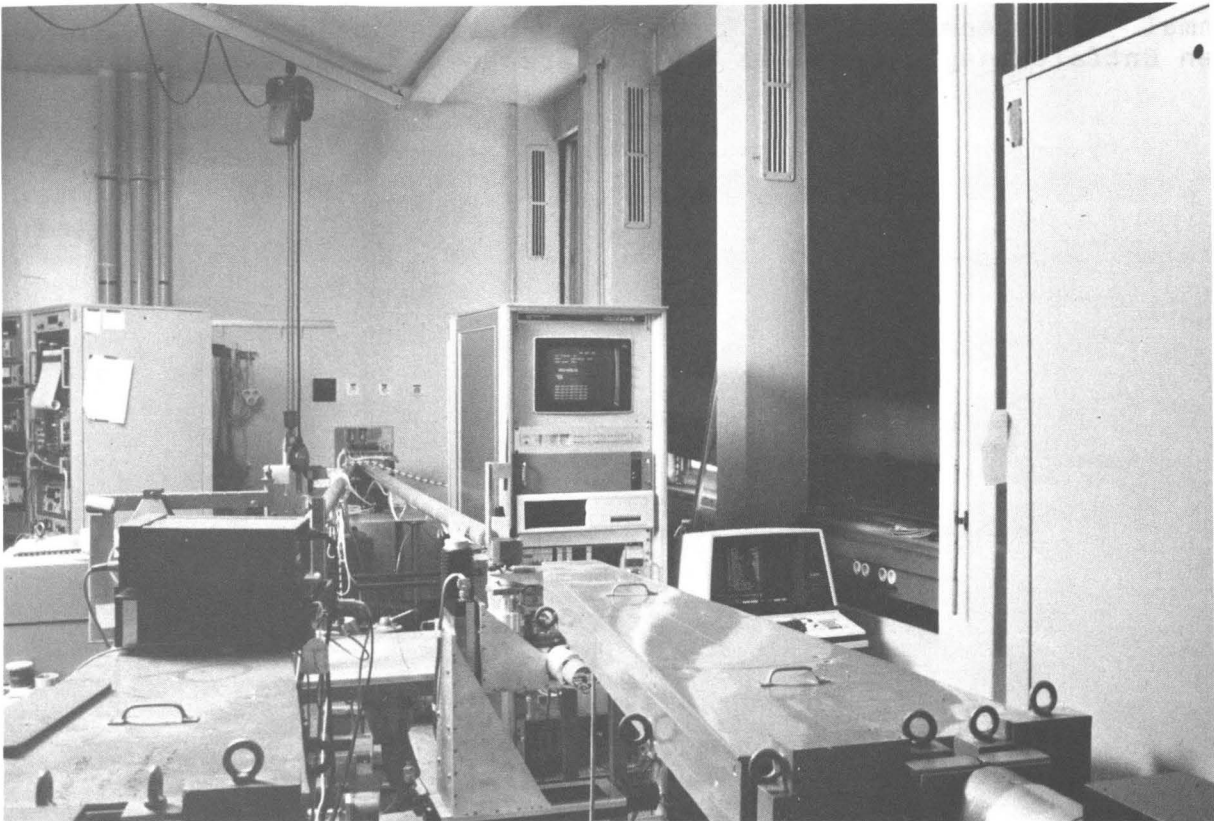


Foto 1: Das Gamma-Diffraktometer

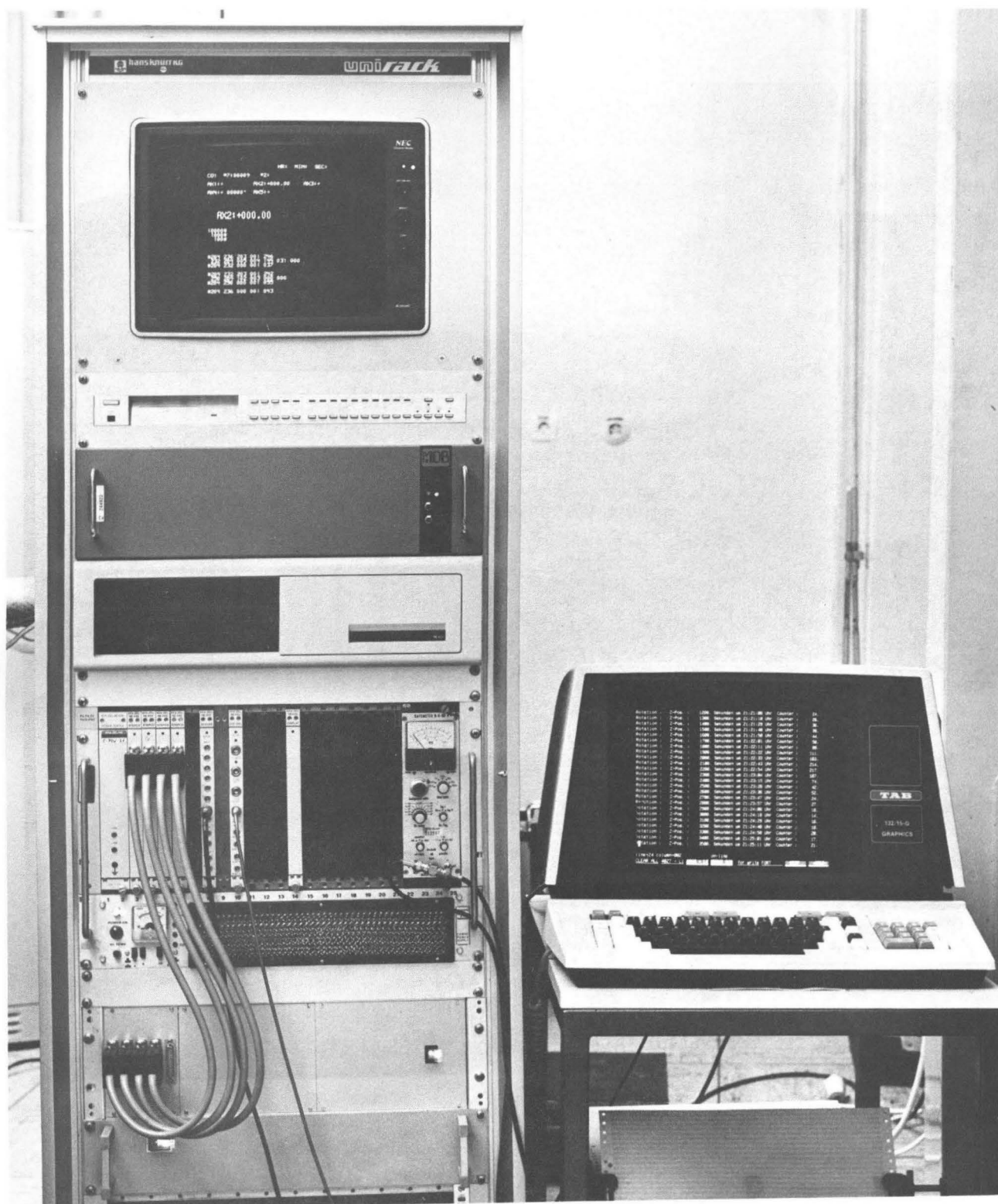


Foto 2: Der Rechner und die Konsole



Foto 3: Ein im Weltraumlabor SPACELAB gezüchteter Silizium-Einkristall wird von Dr.W. Uelhoff auf dem Gamma-Diffraktometer justiert.

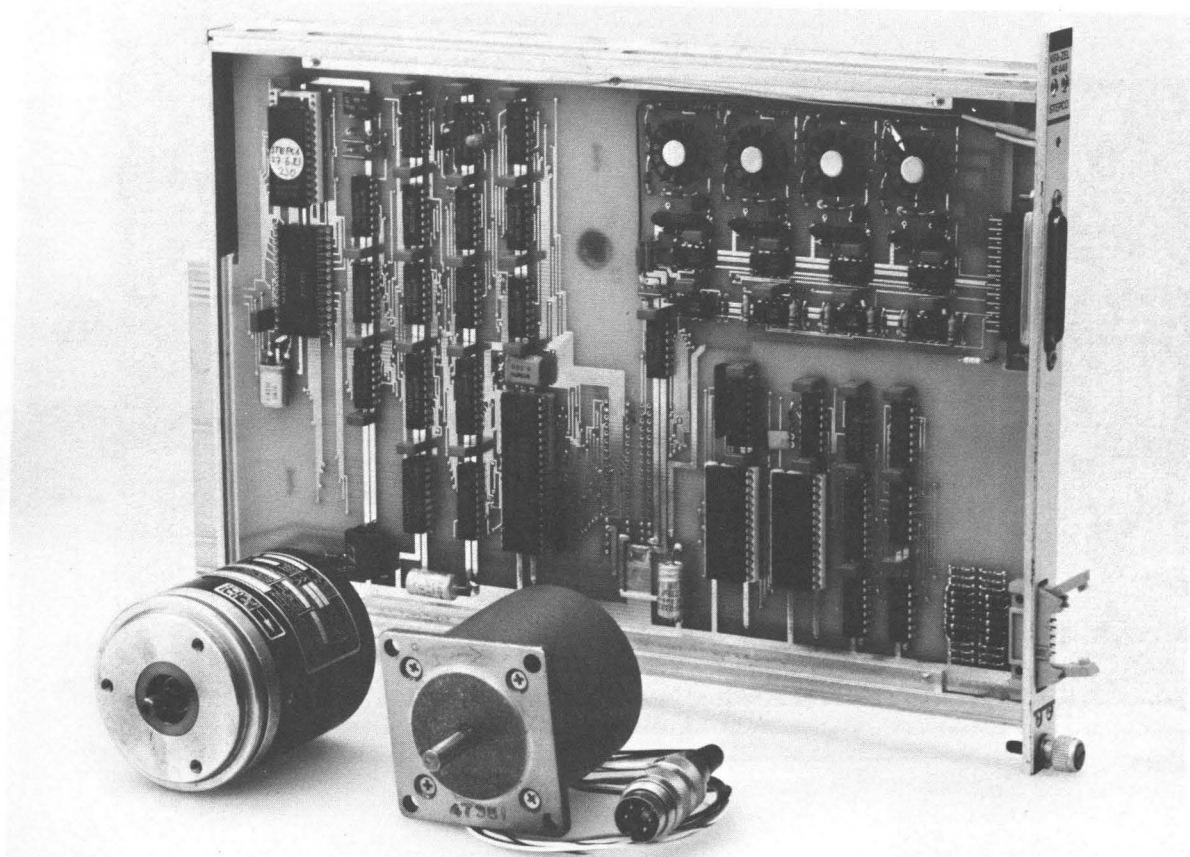


Foto 4: Die Motorsteuerung (I\*\*2 C-Bus)

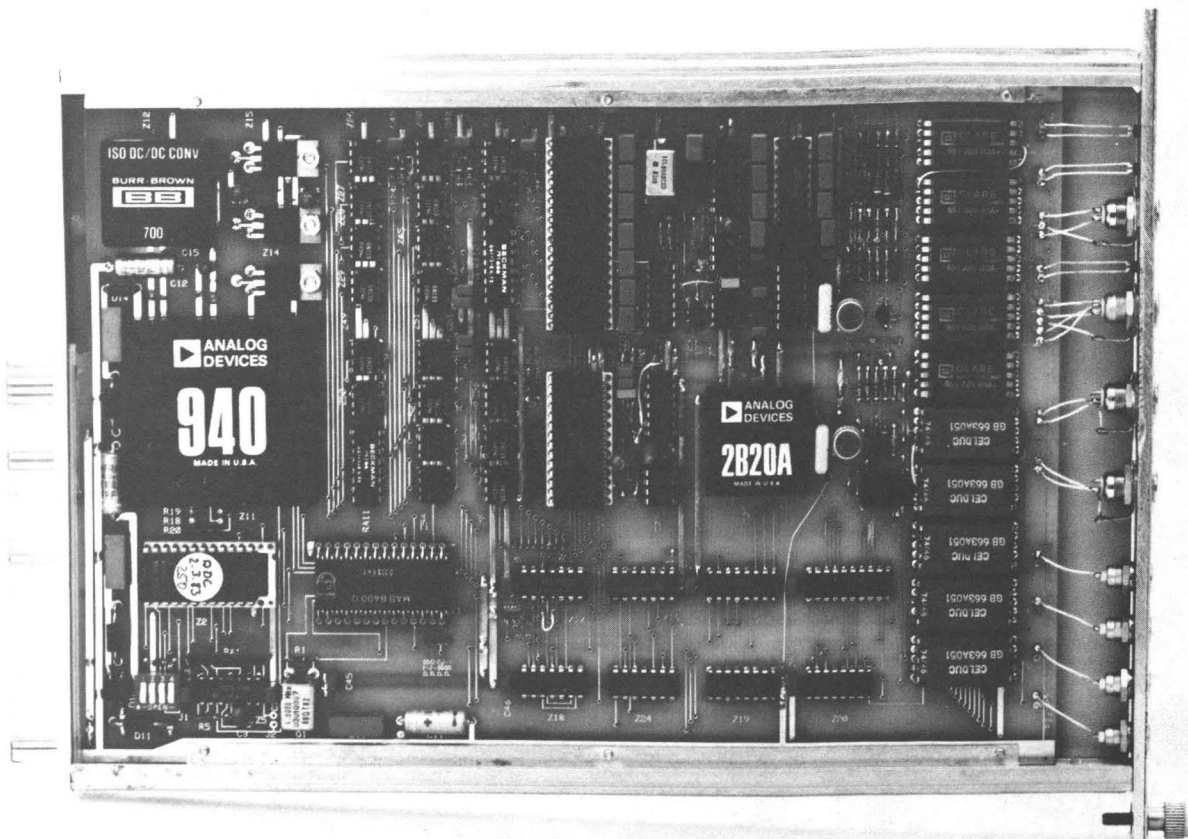


Foto 5: Der ADC/DAC Konverter (I\*\*2 C-Bus)



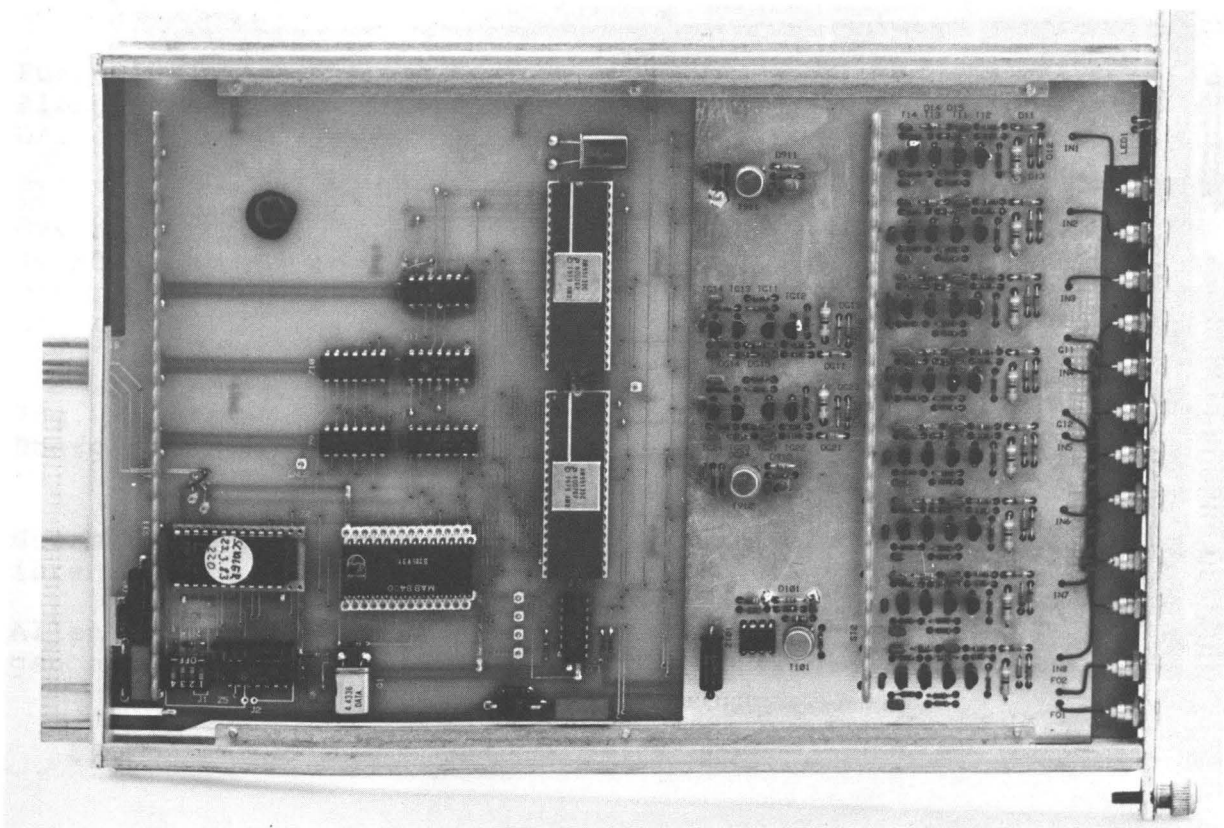


Foto 6: Der Zähler (I\*\*2 C-Bus)

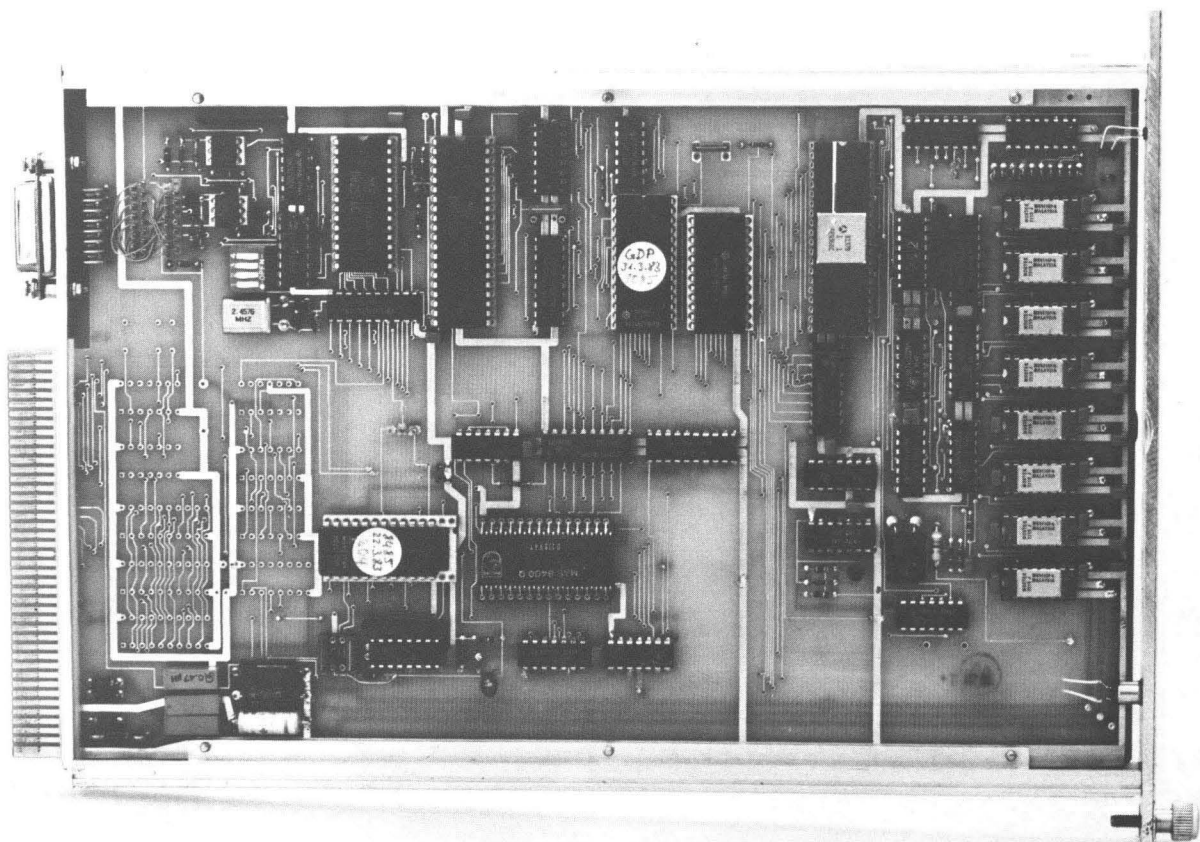


Foto 7: Die Display-Steuerung (I\*\*2 C-Bus)

## Kapitel 17

### DANKSAGUNG

Fuer Anregungen, Hilfe und ständige Diskussionsbereitschaft bei der Planung des Ausbaus danken wir den Herren Prof. Dr. H. Wenzl, Dr. J.-M. Welter und Dr. W. Uelhoff.

Herrn Dr. K.D. Müller, der uns von seiten des ZEL jegliche Hilfe zur Verfügung stellte und Herrn Dr. H. Halling, der uns auch in Detailproblemen immer hilfsbereit zur Seite stand, gebührt ebenfalls unser Dank.

Ing. K. Bickmann gilt unserer Dank für seine Hilfe beim reibungslosen Übergang auf das neue System.

Herrn Dipl.-Phys. F. J. Bremer und Dipl.-Phys. P. Meuffels sei für ihre Unterstützung ebenfalls gedankt.

Allen Mitarbeitern des ZEL und des IFF, deren Hilfe wir in Anspruch genommen haben, sei hiermit unser Dank ausgesprochen.



## Kapitel 18

### LITERATUR

1. J.R. Schneider  
A Gamma-ray diffractometer: A tool for investigating mosaics structure  
J. Appl. Cryst. 7, 541-546 (1974)
2. J.R. Schneider  
Interpretation of rocking curves measured by Gamma-ray diffractometry  
J. Appl. Cryst. 7, 547-554 (1974)
3. G. Mair, H.J. Fenzl, H. Bleichert, L. Gain  
Application of Gamma-ray diffraction in materials science  
Nucl. Instrum. & Meth. 166, 59-63 (1979)
4. J.R. Schneider  
Applications of Gamma-ray diffractometry  
Nuclear Science Applications (S. 227-276)  
Harwood Acad. Publishers GmbH (1981)
5. DEC, Maynard, Massachusetts  
System Generation and Installation Guide  
Order No. AA-M625C-TC
6. DEC, Maynard, Massachusetts  
IAS/RSX-11 I/O Operations Reference Manual  
Order No. AA-M176A-TC

